

# Cryptographic protocols based on inner product spaces and group theory with a special focus on the use of Nielsen transformations

Dissertation  
zur Erlangung des Doktorgrades  
an der Fakultät für  
Mathematik, Informatik und Naturwissenschaften  
Fachbereich Mathematik  
der Universität Hamburg

vorgelegt von

Anja I. S. Moldenhauer

Hamburg, 2016

Tag der Disputation: 23.09.2016

Als Dissertation angenommen vom Fachbereich  
Mathematik der Universität Hamburg

Auf Grund der Gutachten von Prof. Dr. Gerhard Rosenberger

und Prof. Dr. Ulf Kühn

und Prof. Dr. Bettina Eick

To my family / Für meine Familie



# Contents

<b>1. Introduction</b>	<b>9</b>
1.1. Cryptology and cryptographic protocols . . . . .	10
1.2. Examples of some known cryptographic protocols . . . . .	15
1.2.1. Diffie-Hellman key exchange protocol . . . . .	15
1.2.2. ElGamal public key cryptosystem . . . . .	15
1.2.3. Pohlig-Hellman private key cryptosystem . . . . .	16
1.2.4. RSA public key cryptosystem . . . . .	17
1.3. Outline of this thesis and summary of results . . . . .	18
1.3.1. On the evolution of the thesis . . . . .	18
1.3.2. Summary of the chapters and developed cryptographic protocols . . . . .	20
1.3.3. Assessment of the results . . . . .	54
1.4. Suggestions for other platform groups instead of finitely generated free groups . .	56
1.5. Open questions and further research for cryptographic protocols based on combi- natorial group theory . . . . .	59
<b>2. Inner product spaces and cryptography</b>	<b>61</b>
2.1. Inner product spaces and a private key cryptosystem (Protocol 1) . . . . .	63
2.2. Inner product spaces and a challenge and response protocol (Protocol 2) . . . . .	68
<b>3. A group theoretical ElGamal cryptosystem based on a semidirect product of groups</b>	<b>73</b>
3.1. ElGamal like public key cryptosystem (Protocol 3) . . . . .	75
3.2. Signature with a semigroup of $3 \times 3$ matrices over $\mathbb{F}_7[A_5]$ (Protocol 4) . . . . .	81
3.3. Security and ongoing research about the HKKS-key exchange protocol . . . . .	87
<b>4. Combinatorial group theory</b>	<b>93</b>
4.1. Free groups and group presentations . . . . .	93
4.2. Nielsen transformations, Nielsen reduced sets and additional theory . . . . .	96
4.3. Fundamental problems in group theory . . . . .	103
4.4. Whitehead-Automorphisms . . . . .	106
<b>5. Secret sharing protocols</b>	<b>109</b>
5.1. D. Panagopoulos' $(n, t)$ -secret sharing scheme . . . . .	111
5.1.1. Share distribution method given by D. Panagopoulos . . . . .	113
5.2. A purely combinatorial $(n, t)$ -secret sharing scheme (Protocol 5) . . . . .	113
5.3. Access structures for generalized secret sharing schemes . . . . .	118
5.3.1. Generalized secret sharing schemes by M. Ito, A. Saito and T. Nishizeki .	120
5.3.2. Generalized secret sharing schemes by J. Benaloh and J. Leichter . . . . .	125
5.4. Comparison with A. Shamir's suggested properties . . . . .	130
<b>6. Secret sharing schemes using Nielsen transformations</b>	<b>135</b>
6.1. Secret sharing scheme based on Nielsen transformations and $SL(2, \mathbb{Q})$ (Protocol 6)	135
6.2. Secret sharing scheme based on Nielsen reduced sets and the free length (Protocol 7)	146

<b>7. Private key cryptosystem with <math>Aut(F)</math> (Protocol 8)</b>	<b>153</b>
7.1. Modification with the ciphertext a reduced word for the cryptosystem with $Aut(F)$	166
7.2. Modification with $SL(2, \mathbb{Q})$ for the cryptosystem with $Aut(F)$ . . . . .	171
7.3. Modification with Hilbert's Tenth Problem for the cryptosystem with $Aut(F)$ . .	175
7.4. Chosen plaintext attacks on the cryptosystem with $Aut(F)$ . . . . .	180
7.5. Chosen ciphertext attacks on the cryptosystem with $Aut(F)$ . . . . .	182
<b>8. Private key cryptosystem with <math>Aut(F_U)</math> (Protocol 9)</b>	<b>187</b>
8.1. Modification with the ciphertext a reduced word for the cryptosystem with $Aut(F_U)$	200
8.2. Modification with $SL(2, \mathbb{Q})$ for the cryptosystem with $Aut(F_U)$ . . . . .	205
8.3. Modification with Hilbert's Tenth Problem for the cryptosystem with $Aut(F_U)$ .	206
8.4. Chosen plaintext attacks on the cryptosystem with $Aut(F_U)$ . . . . .	208
8.5. Chosen ciphertext attacks on the cryptosystem with $Aut(F_U)$ . . . . .	210
<b>9. Private key cryptosystem which uses automorphisms on plaintext sequences (Protocol 10)</b>	<b>215</b>
9.1. Chosen plaintext attacks on the cryptosystem which uses automorphisms on plaintext sequences . . . . .	234
9.2. Chosen ciphertext attacks on the cryptosystem which uses automorphisms on plaintext sequences . . . . .	234
<b>10. Additional cryptographic protocols using automorphisms of finitely generated free groups</b>	<b>235</b>
10.1. ElGamal like public key cryptosystem using automorphisms on a finitely generated free group $F$ (Protocol 11) . . . . .	235
10.2. Challenge and response protocol using automorphisms on a finitely generated free group $F$ (Protocol 12) . . . . .	239
<b>A. Additional definitions</b>	<b>243</b>
A.1. Boolean formulae . . . . .	243
A.2. Elementary free groups . . . . .	245
<b>B. Additional examples</b>	<b>247</b>
B.1. Example for automorphisms for Remark 7.0.10 . . . . .	247
B.2. A part of an example with additional information from Alice . . . . .	253
B.3. Example for Remark 7.0.9 . . . . .	257
<b>C. Calculations with Maple 16 or GAP for examples</b>	<b>259</b>
C.1. Example 2.1.5 calculations in Maple 16 . . . . .	259
C.2. Example 2.2.2 calculations in Maple 16 . . . . .	262
C.3. Example 6.1.3 calculations in Maple 16 . . . . .	267
C.4. Example 6.2.3 executed with GAP . . . . .	274
C.5. Example 7.0.7 executed with GAP . . . . .	277
C.6. Example 7.2.4 calculations in Maple 16 and GAP . . . . .	287
C.7. Example of a message, where inverse automorphisms were used for decryption in a cryptosystem based on $Aut(F)$ . . . . .	307
C.8. Example 8.0.4 calculated with GAP . . . . .	316
C.9. Example for decryption where Bob uses an algorithm to solve a constructive membership problem for a cryptosystem based on $Aut(F_U)$ . . . . .	323
C.10. Example 9.0.7 calculated with GAP and Maple 16 . . . . .	331
C.11. Example 10.1.4 executed with GAP . . . . .	346

C.12.Example 10.2.2 executed with GAP and Maple 16 . . . . .	351
<b>Bibliography</b>	<b>357</b>





# Chapter 1

## Introduction

At the present time the most widely used cryptographic protocols are based on number theory and on the structure of commutative groups. These include for example RSA, Diffie-Hellman, ElGamal and elliptic curve cryptography (see for instance [Sil09] or [Kna92]).

Due to the growing strength of computers and the increased sophistication of improved computational techniques there is a definite need for research concerning new cryptographic protocols.

For the most used cryptographic protocols, like RSA and Diffie-Hellman, there are two problems, integer factorization and finding discrete logarithms, that provide the security certification. Unfortunately there exist algorithms on a hypothetical quantum computer which factor integers and find discrete logarithms in polynomial time (see [Sho96]<sup>1</sup>). Both problems are hard to solve on classical computers, but if a working quantum computer is developed, in the future the cryptographic protocols based on these problems are no longer secure. Therefore, it is necessary for cryptographic protocols to be modified.

An idea that has been pursued is to use non-commutative groups as cryptographic platforms. One of the earliest suggestions to use those groups was given by W. Magnus and appeared 1973 in his paper [Mag73]. However, it was the pioneering paper [MW85] by M. R. Magyarik and N. R. Wagner in 1985, where they introduced the innovative idea of using the difficulty of group theoretical decision problems (they suggest the word problem) as one-way functions in cryptography, that began the work on using non-commutative platforms.

This led to an active line of research which tries to develop cryptographic protocols based on non-commutative platforms. This area is called **non-commutative algebraic cryptography**. Due to the fact, that the best understood non-commutative platforms are groups, this research area is also known as **group based cryptography**. The book [MSU08] from A. Myasnikov, V. Shpilrain and A. Ushakov contains a good overview of group based cryptography.

In this thesis, we give extensions of known cryptographic protocols, develop new cryptographic protocols and give modifications of cryptographic protocols. The focus lays on newly developed cryptographic protocols using non-commutative groups, and the use of techniques which are typically studied in combinatorial group theory.

With extensions we mean, that the mathematical idea behind a known cryptographic protocol is used to generate another cryptographic protocol based on this theory. For example we use the idea behind the CFRZ-secret sharing scheme to come up with a private key cryptosystem. To develop new cryptographic protocols means, that we use a mathematical theory which was not used before to get a cryptographic protocol in such a way, thus the based theory is new for this cryptographic protocol. For example we use finitely generated free groups, Nielsen reduced

---

<sup>1</sup>A preliminary version of the paper appeared in the proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe 1994, IEEE Computer Society Press, 124–134.

sets and automorphisms on finitely generated free groups, given by Nielsen transformations or Whitehead-Automorphisms, respectively, to develop symmetric key cryptosystems. By a modification we use the mathematical idea behind a cryptographic protocol but add something to it, which, in general, improves the security. For example in addition to a newly developed symmetric key cryptosystem, based on combinatorial group theory, we use a faithful representation of a finitely generated free group  $F$  into  $SL(2, \mathbb{Q})$ .

In progress of writing this thesis the following relating publications arose:

- [FMR13] B. Fine, A. I. S. Moldenhauer and G. Rosenberger, *A secret sharing scheme based on the Closest Vector Theorem and a modification to a private key cryptosystem*, De Gruyter Groups Complexity Cryptology **5** (2013), 223–238.
- [Mol15] A. I. S. Moldenhauer, *A group theoretical ElGamal cryptosystem based on a semidirect product of groups and a proposal for a signature protocol*, Contemporary Mathematics **633** (2015), 97–113.
- [MR15] A. I. S. Moldenhauer and G. Rosenberger, *Cryptographic protocols based on Nielsen transformations*, ArXiv: <https://arxiv.org/abs/1504.03141v1> (2015).
- [FKIMR15] B. Fine, G. Kern-Isberner, A. I. S. Moldenhauer and G. Rosenberger, *On the Generalized Hurwitz Equation and the Baragar-Umeda Equation*, Results in Mathematics **69** (2015), 69–92.
- [MR16] A. I. S. Moldenhauer and G. Rosenberger, *Cryptosystems using automorphisms of finitely generated free groups*, Tributes **29**, Computational Models of Rationality (2016), 31–51.
- [CFMRZ16] C. S. Chum, B. Fine, A. I. S. Moldenhauer, G. Rosenberger and X. Zhang, *On secret sharing protocols*, Contemporary Mathematics (to appear 2016).

In this chapter we first give a brief overview about cryptology and cryptographic protocols. Examples of mostly standard cryptographic protocols are given. Following by the outline of this thesis in which we also give a summary of results for each chapter and sketch the developed cryptographic protocols in tables. Afterwards a section about the assessment of the results of this thesis is given. Suggestions for other platform groups instead of finitely generated free groups for the newly developed cryptographic protocols, based on combinatorial group theory, are explained. The chapter closes with open questions and further research ideas for cryptographic protocols based on combinatorial group theory.

## 1.1. Cryptology and cryptographic protocols

This section gives a brief overview of cryptology. For more information see for instance [MvOV97], [Buc10], [BFKR15] or [BNS10]. For number theory and cryptography see for example [Kob87]. For group theoretical cryptography see for instance [VS15] and especially for group-based cryptography with a special focus on non-commutative groups see for example [MSU08].

Historically, cryptology is an old subject, which started with the need for secrecy in information and messages, respectively. For example Gaius Iulius Caesar (100 B.C - 40 B.C) used

substitutions for military communication purposes (known as Caesar Cipher, see for instance [MvOV97]).

**Cryptology** contains two subfields: cryptography and cryptanalysis. **Cryptography** is the science of developing and implementing cryptographic protocols. **Cryptanalysis** is the science of breaking cryptographic protocols. In cryptanalysis the cryptographic protocols are analyzed and the strengths and weaknesses are presented. In most literatures cryptography is used synonymously with cryptology.

Nowadays, with the use of computers and the internet, there are different goals and purposes for cryptographic methods; these include confidentiality, data integrity, authentication and non-repudiation.

These cryptographic goals given in greater detail (following [MvOV97]) are:

- **Confidentiality** is a technique used to keep the content of information from all but those authorized to have it. **Secrecy** is a term synonymous with confidentiality and privacy. There are numerous approaches to providing confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.
- **Data integrity** is a technique which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion and substitution.
- **Authentication** is a technique related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, data of origin, data content, time sent, etc. For these reasons this aspect of cryptography is usually subdivided into two major classes: **entity authentication** and **data origin authentication**. Data origin authentication implicitly provides data integrity (for if a message is modified, the source has changed).
- **Non-repudiation** is a technique which prevents an entity from denying previous commitment or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary. For example, one entity may authorize the purchase of property by another entity and later deny such authorization was granted. A procedure involving a trusted third party is needed to resolve the dispute.

A **cryptographic task** is where one or more parties have to communicate with some degree of secrecy and would like to get one or more of the above cryptographic aims. Suppose that several parties want to manage a cryptographic task. Then they have to communicate with each other and cooperate. Hence, each party has to follow certain rules and implement certain agreed upon algorithms. The set of all such methods and rules to perform a cryptographic task is called a **cryptographic protocol**.

Cryptology is subjected to **Kerckhoffs' Principle**<sup>2</sup>:

*The security of a cryptosystem must only lie in the choice of its keys; everything else (including the algorithm itself) should be considered public knowledge (see for instance [vTJ11]).*

This principle can be extended to all cryptographic protocols. The algorithms used in cryptographic protocols are public knowledge and security depends only on the secrecy of the private assumptions or keys, respectively, of the parties in a cryptographic protocol.

---

<sup>2</sup>Synonyms: Kerckhoffs' law; Shannon's maxim

The two parties in cryptographic protocols are often referred to as Alice and Bob. We differentiate between the following cryptographic protocols:

- **Cryptosystems:**

A cryptosystem is an algorithm to change an original message, written in some alphabet, also called **plaintext**, into a coded text, also called **ciphertext**, and vice versa. The aim of this cryptographic protocol is to protect the secrecy of the original message, if Alice and Bob communicate with each other over a public channel. The process of putting a plaintext into a ciphertext is called **encryption** while the reverse procedure, that is, putting a ciphertext into the plaintext (more precisely into the original message), is called **decryption**. A formal definition is the following:

**Definition 1.1.1.** A cryptosystem exists of the following parts and property:

- (i) A set  $\mathcal{P}$  of plaintexts.
- (ii) A set  $\mathcal{C}$  of ciphertexts.
- (iii) A set  $\mathcal{K}$  of keys, also called key space.
- (iv) A set  $\mathcal{E} := \{E_e \mid e \in \mathcal{K}\}$ , with mappings

$$E_e : \mathcal{P} \rightarrow \mathcal{C}$$

for encryption. The key  $e$  is called encryption key.

- (v) A set  $\mathcal{D} := \{D_d \mid d \in \mathcal{K}\}$ , with mappings

$$D_d : \mathcal{C} \rightarrow \mathcal{P}$$

for decryption. The key  $d$  is called decryption key.

- (vi) For each  $e \in \mathcal{K}$  exists a  $d \in \mathcal{K}$ , such that  $D_d(E_e(p)) = p$  for all  $p \in \mathcal{P}$ .

There are two different kinds of cryptosystems.

1. **Private key cryptosystems** (also called symmetric (key) cryptosystems):

The encryption key  $e$  is the same key as the decryption key  $d$  or it is easy to determine  $d$  knowing only  $e$  or vice versa. Therefore, Alice and Bob have to agree (privately or with the help of a key exchange or transport protocol) on a common secret encryption key.

2. **Public key cryptosystems** (also called asymmetric (key) cryptosystems):

Given the encryption key  $e$  it is almost infeasible to determine the corresponding decryption key  $d$ . The encryption key  $e$  is a public key. If  $e$  is the public key of Alice everyone including Bob is able to send her a message, but only Alice who knows the corresponding not public decryption key  $d$  to  $e$ , such that  $D_d(E_e(p)) = p$  for all  $p \in \mathcal{P}$ , is able to decrypt the ciphertexts correctly.

The basic idea is that a so called one-way function  $f$  is used to encrypt a plaintext. Informally speaking, a one-way function is a function for which it is easy to compute the image  $f(x)$  for an element  $x$  in the domain of  $f$ , but it is very hard to calculate  $f^{-1}(y)$  for “most”  $y$  in the codomain of  $f$ .

- **Signature protocols (also called digital signature protocols):**

In a signature protocol the receiver of a message or piece of information is able to verify the transmitter authenticity. However, the receiver is not able to masquerade himself as the original transmitter and sends messages to another party in the name of the original transmitter. Signature protocols are able to protect data integrity, authentication or non-repudiation.

- **Authentication protocols:** A member authentication protocol should answer the question if the current communication partner is exactly the person for which he claims to be. The prover must be able to identify himself in real time to a verifier. We differentiate between one-way and two-way authentication depending if only the verifier identifies himself to the prover or if in addition the prover verifies himself also to the verifier.

A special kind of authentication protocols is a **challenge and response protocol**. In a challenge and response protocol there are two participants. One is called the **verifier** and the other one is called the **prover**. The verifier presents a question (“challenge”) to the prover and the prover has to provide the correct answer (“response”). They perform different steps. First, they agree privately on a common shared secret between the prover and the verifier which is a tuple  $(P, V)$  where  $P$  is a standard password for the prover and  $V$  is the associated challenge “space”. The challenge “space” provides an unlimited set of back-up challenges to the password. The assumption is, that it is infeasible to answer the challenges without knowing the challenge “space”. Second, the prover sends the password to the verifier. Third, the verifier presents a question depending on the associated challenge “space”  $V$ . Fourth, the question is answered by the prover and the answer is sent to the verifier. Fifth, the verifier proves if the response is correct. This is then repeated a finite number of questions. If all questions are answered correctly the prover (and the password  $P$ ) is verified.

- **Zero-Knowledge protocols:** Alice shall prove to Bob, that she is aware of a certain secret information, without revealing any information. For example she can use this cryptographic protocol to identify herself to Bob by verifying her knowledge about some special information. Alice wants to convince Bob of the correctness of a claim whereby Bob is allowed to ask questions. We define an interactive Zero-Knowledge proof roughly by the following properties (following [BNS10]):
  - In case that Alice can prove the correctness of her claim she can always convince Bob to believe her (feasibility).
  - In case that Alice’s claim is wrong or she is not able to prove it, there is just a small probability for her to convince Bob (correctness).
  - The proof has the Zero-Knowledge property. This means, the only knowledge Bob gains is that Alice can prove her claim. Formally speaking: there is a simulator which can, without actually knowing a proof, construct an interactive proof, which is for a third party not distinguishable of a real interactive proof.

Zero-Knowledge protocols can be used for authentication.

- **Key exchange protocols and key transport protocols:** If Alice and Bob want to communicate via a private key cryptosystem they use for example a key exchange protocol to exchange a secret private key. A key transport protocol is used, if one party chooses the private key and transmits it to the other party.
- **Secret sharing protocols:** After Kerckhoffs’ Principle the security of a cryptographic protocol lies only in the choice of its secret keys. This provides a strong motivation for the idea of secret sharing protocols. As already mentioned in [BFKR15] if we examine the problem of maintaining sensitive information, we consider two issues: secrecy and availability of the information. If only one person keeps the entire secret (which is for example a key for a cryptographic protocol), then there is a risk that the person might lose the secret or the person might not be available when the secret is needed. Hence, it is often wise to allow several entities to have access to the secret. Otherwise, the more people who can access the secret, the higher the chance the secret will be leaked. A secret sharing

protocol is designed to solve these issues by splitting a secret into multiple shares and distributing these shares among a group of participants. The secret can only be recovered when the participants of an authorized subset join together to combine their shares. In this thesis we mainly work with  $(n, t)$ -secret sharing protocols. A  $(n, t)$ -**secret sharing protocol** (or  $(n, t)$ -threshold scheme with a threshold  $t$ ),  $n, t \in \mathbb{N}$  and  $t \leq n$ , is a method to distribute a secret among a group of  $n$  participants in such a way that it can be recovered only if at least  $t$  of them combine their shares. The person who calculates and distributes the shares to the participant is called **dealer**. For a formal definition and more details see Chapter 5.

We can replace the word “protocol” or “cryptographic protocol” by the word “scheme”.

Symmetric cryptosystems are very old in comparison with asymmetric cryptosystems. The first known military symmetric cryptosystem was the scytale used in the 5th century BC by the Spartans (see for instance [Sin06]). A scytale is a wooden staff around which a stripe of leather or parchment is wound. The message is written on the stripe along the wooden staff. If it is taken off of the wooden staff the stripe appears with an arbitrary looking order of letters. Only a scytale with the same diameter as the first one decipher the message correctly, when the stripe is wrapped around it.

The first asymmetric protocol was a key exchange protocol introduced 1976 by W. Diffie and M. Hellman in [DH76], see also Section 1.2.1. R. L. Rivest, A. Shamir and L. Adleman published 1978 the paper [RSA78] in which they describe the first asymmetric cryptosystem, known as RSA, see also Section 1.2.4.

For more historical background see for example [Kah96] or [Sin06].

An attempt to break a cryptographic protocol is called an **attack**. A brute force attack is an attack where all possible candidates for a key are tested. There are other different attacks on cryptosystems which are, in general, more efficient than brute force attacks, see [BNS10]: Assume an enemy has access to an oracle which is able to perform the cryptosystem in which the enemy is interested. It is like a blackbox. The oracle is able to encrypt and decrypt but it does not tell the enemy which keys it has used for which encryption. The enemy can see at most the plaintext and the ciphertext and, depending on the attack, he can determine which plaintext or ciphertext the oracle decrypts.

- **Known ciphertext attacks:** An oracle chooses random plaintexts and calculates the depending ciphertexts. These ciphertexts are given to the enemy. This is also known as ciphertext only attack, where the enemy gets the ciphertexts which are sent between Alice and Bob.
- **Known plaintext attacks:** An oracle chooses random plaintexts and calculates the depending ciphertexts. The enemy gets both, the plaintexts and the corresponding ciphertexts. Thus, the enemy gets plaintext-ciphertext pairs he did not choose.
- **Chosen plaintext attacks:** The enemy chooses plaintexts and sends these to the oracle. The plaintexts are encrypted by the oracle and sent back to the enemy.
- **Chosen ciphertext attacks:** The enemy chooses ciphertexts and sends these to the oracle. The ciphertexts are then decrypted and the plaintexts are sent back to the enemy.

We analyze the developed private key cryptosystems concerning known ciphertext attacks, because these attacks are always possible. In addition we take a closer look at chosen plaintext attacks and chosen ciphertext attacks. A chosen plaintext attack is stronger than a known plaintext attack, because an eavesdropper, Eve, is able to actively influence for which plaintexts

she gets the corresponding ciphertexts. For example it is useful for Eve to encrypt a plaintext which consists of just the same alphabet letter to get information about the way how this letter is encrypted. For example she sends the plaintext “aaaaaa” to the oracle to get hopefully information on how the letter “a” is encrypted. With chosen plaintext and chosen ciphertext attacks the enemy is able to get plaintext-ciphertext pairs, which he chooses. For us this is of more interest than known plaintext attacks and therefore this kind of attacks is not studied.

## 1.2. Examples of some known cryptographic protocols

We present classical concepts of known cryptographic protocols without a full cryptanalysis. These examples were chosen, because they are mostly standard, we already referred to them above and we will need some of these concepts in this thesis.

### 1.2.1. Diffie-Hellman key exchange protocol

The simplest and original implementation of this cryptographic protocol by W. Diffie and M. E. Hellman (see [DH76] or [MSU08, Section 1.2]) uses the multiplicative group of integers modulo  $p$ , where  $p$  is a prime number. A more general description of this cryptographic protocol uses an arbitrary finite cyclic group.

The **Diffie-Hellman key exchange protocol** is as follows:

1. Alice and Bob agree on a finite cyclic group  $G$  and a generating element  $g$  in  $G$ . The group  $G$  is written multiplicatively.
2. Alice picks a random natural number  $a$  and sends  $g^a$  to Bob.
3. Bob picks a random natural number  $b$  and sends  $g^b$  to Alice.
4. Alice computes  $K_A = (g^b)^a = g^{ba}$ .
5. Bob computes  $K_B = (g^a)^b = g^{ab}$ .

Since  $ab = ba$  (because  $\mathbb{N}$  is commutative), both, Alice and Bob, are now in possession of the same group element  $K = K_A = K_B$ , which can serve as the shared secret key.

This cryptographic protocol is considered secure against eavesdroppers if  $G$  and  $g$  are chosen properly. The eavesdropper, Eve, must solve the **Diffie-Hellman problem** (recover  $g^{ab}$  from  $g^a$  and  $g^b$ ) to obtain the shared secret key. This is currently considered difficult for a “good” choice of parameters (see for example [MvOV97] for details). In addition, solving the **discrete logarithm problem** (recover  $a$  from  $g$  and  $g^a$ ) would solve the Diffie-Hellman problem and breaks this cryptosystem.

### 1.2.2. ElGamal public key cryptosystem

The ElGamal cryptosystem (see [ElG85] or [MSU08, Section 1.3]) is a public key cryptosystem, which is based on the Diffie-Hellman key exchange protocol (see Section 1.2.1).

The **ElGamal public key cryptosystem** is as follows:

1. Alice and Bob agree on a finite cyclic group  $G$  and a generating element  $g \in G$ .
2. Alice (the receiver) picks a random natural number  $a$  and publishes the element  $c := g^a$ .

3. Bob, who wants to send a message  $m \in G$  to Alice, picks a random natural number  $b$  and sends two elements,  $m \cdot c^b$  and  $g^b$ , to Alice. Note that  $c^b = g^{ab}$ .
4. Alice recovers  $m = (m \cdot c^b) \cdot ((g^b)^a)^{-1}$ .

Because this cryptographic protocol is based on the Diffie-Hellman key exchange protocol the security depends also on the Diffie-Hellman problem and hence it is also vulnerable to the discrete logarithm problem.

### 1.2.3. Pohlig-Hellman private key cryptosystem

The Pohlig-Hellman cryptosystem published in [PH78] is a private key cryptosystem, which is based on number theory more precisely on modular arithmetic and Fermat's little theorem, see for instance [Kob87].

**Theorem 1.2.1.** [Kob87] Fermat's little theorem

*Let  $p$  be a prime number. Any integer  $a$  satisfies  $a^p \equiv a \pmod{p}$ , and any integer  $a$  not divisible by  $p$  satisfies*

$$a^{p-1} \equiv 1 \pmod{p}.$$

Consequently, it is

$$z^x \equiv z^{x \pmod{p-1}} \pmod{p}$$

for  $1 \leq z \leq p-1$  and  $x \in \mathbb{N}$ .

We now describe the **Pohlig-Hellman cryptosystem** and assume that Alice would like to send a message to Bob:

1. Alice and Bob agree on a (large) prime number  $p$ . They choose a natural number  $e$  with  $2 \leq e \leq p-2$  and  $\gcd(e, p-1) = 1$ , and determine  $d$  with  $ed \equiv 1 \pmod{p-1}$ , the inverse element of  $e$  modulo  $p-1$  with  $2 \leq d \leq p-2$ . To calculate  $d$  they can use the Euclidean Algorithm (see for instance [Kob87]).  
Alice stores her private key  $e$  and Bob stores  $d$ .

2. If Alice wants to send a message  $m \in (\mathbb{Z}/p\mathbb{Z})^\times$ , an element of the multiplicative group of  $\mathbb{Z}/p\mathbb{Z}$ , to Bob she calculates

$$c := m^e \pmod{p}$$

and transmits the ciphertext  $c$  to Bob.

3. Bob gets  $c$  and reconstructs the message  $m$  as follows

$$c^d = m^{ed} \equiv m^{ed \pmod{p-1}} \equiv m \pmod{p}$$

because of the consequence of Fermat's little theorem (Theorem 1.2.1) and the choice of  $e$  and  $d$  with  $ed \equiv 1 \pmod{p-1}$ .

Assume an eavesdropper, Eve, gets a ciphertext and the corresponding plaintext, that is, the tuple  $(m, m^e)$ , with  $m \in (\mathbb{Z}/p\mathbb{Z})^\times$ . To break the cryptosystem she tries to get the number  $e$ , because if she knows  $e$  she can calculate  $d$  with the extended Euclidean Algorithm. Getting the number  $e$  of the tuple  $(m, m^e)$  is exactly the discrete logarithm problem.



### 1.2.4. RSA public key cryptosystem

The RSA cryptosystem (see [RSA78] or for instance [Kob87]) can be seen as an extension of the Pohlig-Hellman cryptosystem. However, the RSA cryptosystem is a public key cryptosystem while the Pohlig-Hellman cryptosystem is a private key cryptosystem.

**Definition 1.2.2.** [Kob87]

Let  $n$  be a natural number. The Euler phi-function  $\varphi(n)$  is defined to be the number of non-negative integers  $b$  less than or equal  $n$  which are prime to  $n$ , that is,

$$\varphi(n) := |\{0 \leq b \leq n \mid \gcd(b, n) = 1\}|.$$

**Corollary 1.2.3.** [Kob87]

- The Euler phi-function is “multiplicative”, meaning that

$$\varphi(nm) = \varphi(n)\varphi(m),$$

whenever  $\gcd(n, m) = 1$ .

- If  $p$  is a prime number, then  $\varphi(p) = p - 1$ .

A generalization of Fermat’s little theorem, due to Euler, is used.

**Proposition 1.2.4.** [Kob87]

If  $\gcd(a, n) = 1$ , then  $a^{\varphi(n)} \equiv 1 \pmod{n}$ .

Assume that Alice and Bob would like to communicate with each other and Bob should send a message to Alice, then the **RSA public key cryptosystem** is as follows:

1. Alice chooses privately two prime numbers  $p$  and  $q$ . She calculates  $N := pq$  and  $\varphi(N) = (p - 1)(q - 1)$  (see Corollary 1.2.3). Additionally she needs a number  $e$  with  $2 \leq e \leq \varphi(N) - 1$  and  $\gcd(e, \varphi(N)) = 1$ . The tuple  $(N, e)$  is the public key. She calculates her private key  $d$ , which is her decryption key, by determining  $d$  with

$$ed \equiv 1 \pmod{\varphi(N)}.$$

2. Bob knows the public key  $(N, e)$ . He chooses a plaintext  $m \in (\mathbb{Z}/N\mathbb{Z})^\times$  and computes the ciphertext

$$c \equiv m^e \pmod{N}.$$

Bob sends  $c$  to Alice.

3. Alice, with her decryption key  $d$ , gets the message  $m$  by computing

$$c^d = m^{ed} \equiv m^{ed \pmod{\varphi(N)}} \equiv m \pmod{N},$$

because of Proposition 1.2.4 and the choice of  $e$  and  $d$  with  $ed \equiv 1 \pmod{\varphi(N)}$ .

The numbers  $N = pq$  and  $e$  are public. If an attacker is able to calculate  $\varphi(N) = (p - 1)(q - 1)$  he can generate the multiplicative inverse element of  $e$  in  $\mathbb{Z}/\varphi(N)\mathbb{Z}$  with the Euclidean Algorithm, and hence the description key  $d$ . It is easy to calculate  $\varphi(N)$  if the prime factorization of  $N$  is known. Therefore, the security depends on the factorization problem.

### 1.3. Outline of this thesis and summary of results

We now give the outline of this thesis. Firstly, we explain the evolution of this thesis without going into details of the mathematical theory or the cryptographic protocols. After each paragraph we give a listing of the developed cryptographic protocols. Secondly, we give a summary of each chapter in which we sketch the results and summarize the developed cryptographic protocols. This section closes with an assessment of the results.

#### 1.3.1. On the evolution of the thesis

This Ph.D. project started with extending existing cryptographic protocols to other cryptographic protocols. The basis are the secret sharing schemes, because they were studied in the master's thesis [Mol12] in which different secret sharing protocols were analyzed. Both the CFRZ-secret sharing scheme by C. S. Chum, B. Fine, G. Rosenberger and X. Zhang [CFRZ12], which is based on the Closest Vector Theorem in a real inner product space, and a secret sharing scheme by D. Panagopoulos [Pan10] were analyzed. An observation for both cryptographic protocols is that it is possible to first calculate and distribute the shares for the participants and as a second step to choose the secret (under certain restrictions at the CFRZ-secret sharing scheme) and either send an additional element to each participant or publish this element. With the additional element and the shares the participants are able to reconstruct the secret. The first private key cryptosystem (**Protocol 1**) in this Ph.D. project (published in [FMR13]), is extended from the CFRZ-secret sharing scheme and uses the possibility that the secret can be chosen after the distribution of the shares. In addition a challenge and response system arose a little bit later (**Protocol 2**).

- CFRZ-secret sharing scheme ([CFRZ12], [Mol12])
  - **Protocol 1**: CV-private key cryptosystem ([FMR13])
  - **Protocol 2**: CV-challenge and response protocol

The aim for this thesis is to generate cryptographic protocols in combinatorial group theory, which use non-commutative groups. Thus, the HKKS-key exchange protocol (shorter HKKS-scheme) by M. Habbeb, D. Kahrobaei, C. Koupparis and V. Shpilrain in [HKKS13] was studied, which is a key exchange protocol using semidirect products of (semi)groups. These studies resulted in a public key cryptosystem (ElGamal like) (**Protocol 3**) as well as a signature protocol (**Protocol 4**), published in the paper [Mol15]. There is an ongoing research about the HKKS-scheme with linear algebra attacks and research about suitable platforms, which also affects this cryptosystem and signature protocol. An overview of this research will be given.

- HKKS-key exchange protocol ([HKKS13])
  - **Protocol 3**: Group theoretical ElGamal like public key cryptosystem using semidirect products ([Mol15])
  - **Protocol 4**: Signature with a semigroup of  $3 \times 3$  matrices over  $\mathbb{F}_7[A_5]$  ([Mol15])

Another cryptographic protocol, which is based on the combinatorial group theory and studied in the master's thesis [Mol12], is the secret sharing scheme by D. Panagopoulos [Pan10]. It

uses the word problem in a finitely generated group  $G$  and the shares are subsets of the defining relators of this group. The security depends on the way how the shares are distributed. Based on this observation it was sufficient to use the idea of the share distribution method which D. Panagopoulos describes to get a purely combinatorial  $(n, t)$ -secret sharing scheme (**Protocol 5**). It will be shown that the share distribution method given by D. Panagopoulos is a special case of a multiple assignment scheme introduced in the paper [ISN87] by M. Ito, A. Saito and T. Nishizeki. Furthermore, the introduced combinatorial secret sharing protocol is shown to be similar to a variation of a secret sharing protocol explained in [BL90] by J. Benaloh and J. Leichter. This newly developed secret sharing protocol is published in the survey article [CFMRZ16] as research in the field of secret sharing schemes. It is also published in [MR15].

- D. Panagopoulos' secret sharing scheme ([Pan10], [Mol12])
  - **Protocol 5**: Purely combinatorial  $(n, t)$ -secret sharing scheme ([MR15],[CFMRZ16])

As mentioned above, D. Panagopoulos uses the word problem in finitely generated groups. Therefore, it is possible to calculate first the shares and distribute them to the participants and afterwards determine the secret. The share distribution method can be seen as a basis to develop new cryptographic protocols in this thesis. The aim is to get cryptographic protocols which use combinatorial group theory, hence, a first development is a secret sharing scheme using a finitely generated abstract free group  $F$ , a finitely generated free group in  $\text{SL}(2, \mathbb{Q})$  and Nielsen reduction theory (using Nielsen transformations) (**Protocol 6**). Another secret sharing scheme uses a Nielsen reduced set  $U$  and a Nielsen equivalent set  $V$  to  $U$  (**Protocol 7**). Both cryptographic protocols can be used as  $(n, t)$ -secret sharing schemes using the share distribution method given by D. Panagopoulos. Together with **Protocol 5** the newly developed secret sharing protocols, **Protocol 6** and **Protocol 7**, are published in the survey article [CFMRZ16] as research in the field of secret sharing schemes. They are also published in [MR15].

- D. Panagopoulos' secret sharing scheme ([Pan10], [Mol12]) and Nielsen transformations
  - **Protocol 6**: Secret sharing scheme using Nielsen transformations and  $\text{SL}(2, \mathbb{Q})$  ([MR15], [CFMRZ16])
  - **Protocol 7**: Secret sharing scheme using Nielsen transformations together with Nielsen reduced sets and free lengths of certain words ([MR15], [CFMRZ16])

The studies of these secret sharing protocols with Nielsen transformations can be seen as a basis for the newly developed cryptographic protocols based on combinatorial group theory and Nielsen transformations, which are the main result in this thesis (**Protocol 8** to **Protocol 12**). It was possible to develop two new private key cryptosystems with similar modifications (**Protocol 8** and **Protocol 9**), another new private key cryptosystem (**Protocol 10**), a new ElGamal like public key cryptosystem (**Protocol 11**) and a new challenge and response system (**Protocol 12**), which all use combinatorial group theory and automorphisms on finitely generated free groups. Parts of some of these cryptographic protocols (more precisely parts of **Protocol 8** and **Protocol 11**) are published in [MR16]. **Protocol 11** is also published in [MR15].

- Nielsen transformations

- **Protocol 8:** Private key cryptosystem with  $Aut(F)$  ([MR16])

Modifications:

1. The ciphertext is one reduced word in  $X$
2. The ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$  ([MR16])
3. Hilbert's Tenth Problem is used

- **Protocol 9:** Private key cryptosystem with  $Aut(F_U)$

Modifications:

1. The ciphertext is one reduced word in  $X$
2. The ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$
3. Hilbert's Tenth Problem is used

- **Protocol 10:** Private key cryptosystem using automorphisms on plaintext sequences

- **Protocol 11:** ElGamal like public key cryptosystem using automorphisms on a finitely generated free group  $F$  ([MR15], [MR16])

- **Protocol 12:** Challenge and response protocol using automorphisms on finitely generated free groups

### 1.3.2. Summary of the chapters and developed cryptographic protocols

#### Chapter 2

In Chapter 2 we introduce **Protocol 1** and **Protocol 2**, which extend the CFRZ-secret sharing scheme to a private key cryptosystem as well as to a challenge and response system. The CFRZ-scheme is a  $(n, t)$ -secret sharing protocol, which is based on the Closest Vector Theorem in a real inner product space (see Theorem 1.3.1). The idea behind the CFRZ-scheme was first published by C. S. Chum, B. Fine, G. Rosenberger and X. Zhang in [CFRZ12]. It was worked out and analyzed in detail in [Mol12] whereby parts of these results were published in [FMR13] and the overview article [CFMRZ16].

We require knowledge of linear algebra and analytic geometry, as it is presented for example in the books [Bos08] or [Fis10].

Both cryptographic protocols are based on the following Theorem.

**Theorem 1.3.1.** [Atk89] Closest Vector Theorem

*Let  $W$  be a real inner product space and let  $V$  be a subspace of finite dimension  $t$ ,  $t \in \mathbb{N}$ . Suppose that  $w^* \in W$ , with  $w^* \notin V$ , and  $e_1, e_2, \dots, e_t$  is an orthonormal basis of  $V$ . Then the unique vector  $w \in V$  closest to  $w^*$  is given by*

$$w = \langle w^*, e_1 \rangle e_1 + \langle w^*, e_2 \rangle e_2 + \dots + \langle w^*, e_t \rangle e_t,$$

where  $\langle \cdot, \cdot \rangle$  is the inner product on  $W$ .

**Inner structure of Chapter 2:**

First of all the Closest Vector Theorem and the CFRZ-scheme are recalled. Afterwards **Protocol 1** and **Protocol 2** are introduced in detail and we also give an example for **Protocol 1** and an example for **Protocol 2**.

- **Protocol 1: CV-private key cryptosystem**

**Protocol 1** makes use of the **C**losest **V**ector **T**heorem in a real inner product space, thus we call it CV-private key cryptosystem. It is published in [FMR13].

If  $W$  and  $V$  are given as in Theorem 1.3.1 then it is easy to calculate to each element  $w^* \in W \setminus V$  the closest element  $w$  in  $V$  to the element  $w^*$ . A situation where  $W = \mathbb{R}^3$  and  $V$  is a 2-dimensional subspace of  $W$  (visualized by the yellow area) is given in Figure 1.1.

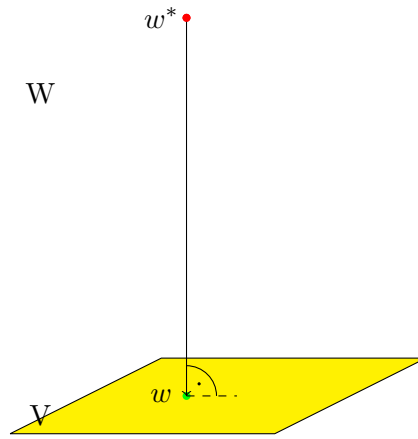


Figure 1.1.: Visualization of a situation with  $W = \mathbb{R}^3$  and  $V$  a 2-dimensional subspace of  $W$

**Protocol 1** is summarized in Table 1.1 (page 22), which is exactly Table 2.1 (page 65) in Chapter 2.

Table 1.1.: Summary of **Protocol 1**: CV-private key cryptosystem

<b>Private Parameters</b>	
A subspace $V \subset W$ with $\dim(V) = t < m$ of a real inner product space $W$ with $\dim(W) = m$ .	
Alice	Bob
Key Creation	
Calculate an orthonormal basis $G = \{e_1, e_2, \dots, e_t\}$ for $V$ .	Calculate the orthogonal complement $V^\perp$ to $V$ and a basis $B^\perp = \{u_1^\perp, u_2^\perp, \dots, u_{m-t}^\perp\}$ for $V^\perp$ .
Encryption	
	Choose plaintext $p \in W$ . Choose arbitrary ephemeral vector $w \in V$ , with $w \neq p$ , and calculate $v := w - p$ . Compute $w^* \in W \setminus V$ : $w^* = \underbrace{w}_{\in V} + \underbrace{(\alpha_1 u_1^\perp + \alpha_2 u_2^\perp + \dots + \alpha_{m-t} u_{m-t}^\perp)}_{=: w^\perp \in V^\perp},$ $\alpha_i \in \mathbb{R}$ and at least one $\alpha_i \neq 0, 1 \leq i \leq m-t$ . Send $c := (w^*, v)$ to Alice. $c := (w^*, v)$
←	
Decryption	
Compute $w = \langle w^*, e_1 \rangle e_1 + \langle w^*, e_2 \rangle e_2 + \dots + \langle w^*, e_t \rangle e_t$ and the message is $w - v = p$ .	

**Security:**

**Protocol 1** is secure against known ciphertext attacks and statistical frequency attack (see for instance [BFKR15]). To be secure against chosen ciphertext and chosen plaintext attacks Alice and Bob have to change the subspace  $V$ , with  $\dim(V) = t$ , after  $t$  encryptions and decryptions, respectively.

**Conclusion:**

Extending a given cryptographic protocol to another cryptographic protocol is interesting, it gives more possibilities to use the suggested mathematical theory for cryptology. Theoretically it is a realizable private key cryptosystem but for applications it has the disadvantage that Alice and Bob have to change the subspace  $V$  not later than after  $t$  messages (in order not to lose privacy).

• **Protocol 2:** CV-challenge and response protocol

**Protocol 2** uses also the Closest Vector Theorem (Theorem 1.3.1).

The verifier and the prover agree on a common secret  $P$  and a corresponding challenge space  $V$ , which is a subspace of a real inner product space  $W$ . After presenting the password  $P$  to the verifier, he gives challenges to the prover, which are correctly solvable in the challenge space  $V$ .

**Possible Challenges:**

We propose two kinds of questions for the challenges.

1. How long is the “line” between  $\ell \geq 3$  associated vectors  $v_1, v_2, \dots, v_\ell \in V$  given the vectors  $v_1^*, v_2^*, \dots, v_\ell^* \in W \setminus V$ . That means, calculate

$$R := \sum_{i=1}^{\ell-1} \|v_i - v_{i+1}\| + \|v_1 - v_\ell\|,$$

whereby  $\|\cdot\|$  denotes the euclidean norm in  $W$ .

2. What is the sum of the entries of the associated vector  $v \in V$  given  $v^* \in W \setminus V$ ?

The situation for a question of case 1. is shown in Figure 1.2, that is, given the elements  $v_1^*, v_2^*$  and  $v_3^*$  it is asks after the length of the blue dotted line.

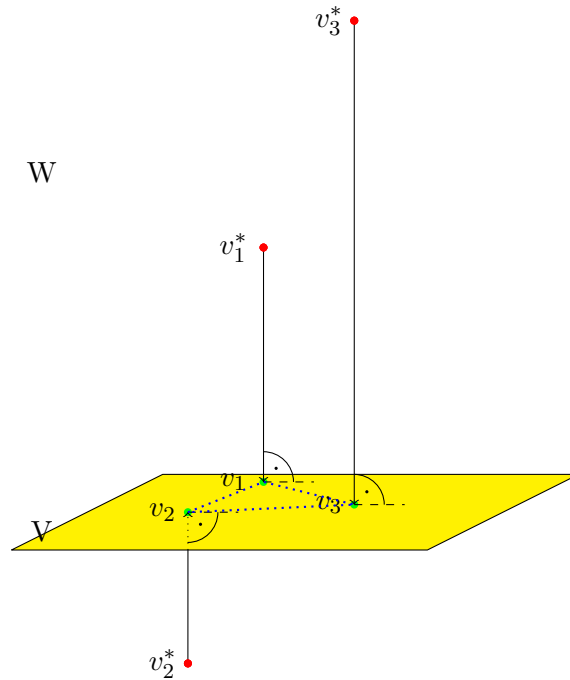


Figure 1.2.: Visualization of a situation in a challenge and response system with  $W = \mathbb{R}^3$  and  $V$  a two dimensional subspace, visualized by a yellow area

**Protocol 2** is summarized in Table 1.2 (page 24), which is exactly Table 2.2 (page 71) in Chapter 2.

Table 1.2.: Summary of **Protocol 2**: CV-challenge and response protocol

<b>Private Parameters</b>	
Subspace $V \subset W$ , with $\dim(V) = t < m$ , of a real inner product space $W$ , with $\dim(W) = m$ , and a common password $P$ . The shared secret is $(P, V)$ .	
Verifier	Prover
Calculate orthogonal complement $V^\perp$ to $V$ and a basis $B^\perp = \{u_1^\perp, u_2^\perp, \dots, u_{m-t}^\perp\}$ for $V^\perp$ .	Calculate orthonormal basis $G = \{e_1, e_2, \dots, e_t\}$ for $V$ .
$\xleftarrow{P}$	
Take challenge space $V$ corresponding to password $P$ , more precisely the calculated orthogonal basis $B^\perp$ . Choose elements $v_1, v_2, \dots, v_\ell \in V$ with $\ell \geq 3$ and calculate the associated elements $v_1^*, v_2^*, \dots, v_\ell^* \in W \setminus V$ . Compute $v_i^* \in W \setminus V$ , $1 \leq i \leq \ell$ : $v_i^* = \underbrace{v_i}_{\in V} + \underbrace{(\alpha_{i_1} u_1^\perp + \alpha_{i_2} u_2^\perp + \dots + \alpha_{i_{m-t}} u_{m-t}^\perp)}_{=: v^\perp \in V^\perp},$ $\alpha_{i_j} \in \mathbb{R}$ and at least one $\alpha_{i_j} \neq 0$ , $1 \leq j \leq m-t$ . Send $v_1^*, v_2^*, \dots, v_\ell^*$ as challenge to the prover.	$\xrightarrow{\text{Challenge: } v_1^*, v_2^*, \dots, v_\ell^*}$
Calculate $R' := \sum_{i=1}^{\ell-1} \ v_i - v_{i+1}\  + \ v_1 - v_\ell\ .$	Compute $v_i = \langle v_i^*, e_1 \rangle e_1 + \langle v_i^*, e_2 \rangle e_2 + \dots + \langle v_i^*, e_t \rangle e_t$ for each $v_i^*$ , $1 \leq i \leq \ell$ . Calculate the response $R$ and send it to the verifier, it is $R := \sum_{i=1}^{\ell-1} \ v_i - v_{i+1}\  + \ v_1 - v_\ell\ .$
$\xleftarrow{\text{Response: } R}$	
Proof if $R' = R$ .	

**Variation:**

It is possible to get a two-way authentication protocol with this challenge and response system. That means the prover authenticates the verifier in the time where the verifier authenticates the prover.



**Security:**

There are infinitely many numbers of possible challenges, thus no challenge is used twice by the verifier. An eavesdropper, Eve, gets only the challenges and the corresponding responses, but these provide not enough information to get the challenge subspace  $V$ . Hence, Eve is not able to masquerade herself as the prover.

**Conclusion:**

Extending the CFRZ-scheme to a challenge and response system is another application of the Closest Vector Theorem for cryptology. It is good to have different challenge “spaces” for challenge and response systems, in particular if these “spaces” generate an infinite amount of challenges as in this CV-challenge and response system. In addition it is a benefit, that it can be used as a two-way authentication protocol.

**Further research questions for Chapter 2:**

We give some ideas for further research questions.

- Are there other cryptographic protocols which can be based on the Closest Vector Theorem, for example a public key cryptosystem or a key exchange protocol?
- Are there other suitable challenges for a challenge and response system using the Closest Vector Theorem?

**Chapter 3**

In Chapter 3 we introduce **Protocol 3** and **Protocol 4**, which extend the HKKS-key exchange protocol (short HKKS-scheme) to an ElGamal like public key cryptosystem as well as to a signature protocol. The HKKS-scheme is introduced by M. Habbeeb, D. Kahrobaei, C. Koupparis and V. Shpilrain in [HKKS13] and it is based on a semidirect product of (semi)groups. **Protocol 3** and **Protocol 4** are published in [Mol15].

First we review some needed background of algebra as it is given in [Rot95] and we require knowledge of algebra, as it is presented for example in the book [JS06].

Let  $G$  and  $H$  be two groups, let  $Aut(G)$  be the group of automorphisms of  $G$  and let  $\rho : H \rightarrow Aut(G)$  be a homomorphism. Then the semidirect product of  $G$  and  $H$  is the set

$$\Gamma = G \rtimes_{\rho} H = \{(g, h) \mid g \in G, h \in H\}$$

with the group operation given by

$$(g, h) \cdot (g', h') = (g^{\rho(h')} \cdot g', h \cdot h').$$

Here  $g^{\rho(h')}$  denotes the image of  $g$  under the automorphism  $\rho(h')$ .

One special case of the semidirect product construction is where the group  $H$  is a subgroup of the group  $Aut(G)$ . If  $H = Aut(G)$ , then the corresponding semidirect product is called the **holomorph** of the group  $G$ . Thus, the holomorph of  $G$ , usually denoted by  $Hol(G)$ , is the set

$$Hol(G) = \{(g, \phi) \mid g \in G, \phi \in Aut(G)\}$$

with the group operation given by

$$(g, \phi) \cdot (g', \phi') = (\phi'(g) \cdot g', \phi \cdot \phi').$$

A product  $\phi \cdot \phi'$  of two homomorphisms means that  $\phi$  is applied first. It is often more practical to use a subgroup of  $Aut(G)$  in this construction.

Let  $G$  be a (semi)group. An element  $g \in G$  as well as an arbitrary automorphism  $\phi \in Aut(G)$  (or an arbitrary endomorphism  $\phi \in End(G)$ ) are chosen and published.

Both, Alice and Bob, are going to work with elements of the form  $(g, \phi^r)$ , where  $g \in G$  and  $r \in \mathbb{N}$ . Note that two elements of this form are multiplied as follows:

$$(g, \phi^r) \cdot (h, \phi^s) = (\phi^s(g) \cdot h, \phi^{r+s}).$$

### Inner structure of Chapter 3:

We first recall the definition of a semidirect product and explain the HKKS-scheme. Next, **Protocol 3**, a public key cryptosystem, is introduced, which is an ElGamal like cryptosystem and is based on a semidirect product of groups. We discuss two platform examples for **Protocol 3**. The first one uses  $\mathbb{F}_p^*$ , whereby  $p$  is a prime number and we denote with  $\mathbb{F}_p$  the field with  $p$  elements and  $\mathbb{F}_p^*$  denotes the multiplicative subgroup. For the endomorphism  $\phi$  of the group  $\mathbb{F}_p^*$  a number  $\ell \in \mathbb{N}$ ,  $\ell > 1$ , is selected, such that

$$\phi(h) = h^\ell \quad \text{for every } h \in \mathbb{F}_p^*.$$

Here, more precisely, it is  $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^*$ .

For the second example we choose a non-commutative group  $G$ , **not** a semigroup, because the inverse of an element  $g^{-k-n}h^{k+n}$ , with  $g, h \in G$ , is needed. For example the general linear group of  $r \times r$  matrices with entries from a field is used, that is,  $G = GL(r, \mathbb{K})$ , with  $r \in \mathbb{N}$ ,  $r > 1$ , and a field  $\mathbb{K}$ . Alice and Bob can use any non-commutative group  $G$  if  $\rho_H$  is selected to be a non-trivial inner automorphism, that is, a conjugation by an element which is not in the center of  $G$ . For any Matrix  $M \in G$  and for any  $k \in \mathbb{N}$ ,  $k > 0$ , it is

$$\rho_H(M) = H^{-1}MH \quad \text{and} \quad \rho_H^k(M) = H^{-k}MH^k.$$

Afterwards, the second cryptographic protocol in this chapter is introduced. **Protocol 4** is a signature scheme with a semigroup of  $3 \times 3$  matrices over  $\mathbb{F}_7[A_5]$ , whereby  $\mathbb{F}_7$  is the field with seven elements and  $A_5$  is the group of even permutations on five symbols. Here, more precisely, it is  $\mathbb{F}_7^* = (\mathbb{Z}/7\mathbb{Z})^*$ .

There is an ongoing research about the HKKS-key exchange protocol, which also affects the cryptographic protocols in this chapter. Therefore, we close Chapter 3 with an overview of this research.

### • **Protocol 3:** Group theoretical ElGamal like public key cryptosystem using semidirect products

We summarize **Protocol 3**, which is a group theoretical ElGamal like public key cryptosystem using semidirect products, in Table 1.3 (page 27), which is exactly Table 3.1 (page 76) in Chapter 3.

### Security:

The security depends on the platform group  $G$  and the automorphism  $\phi \in Aut(G)$ , which are used for this cryptographic protocol. Thus, if the group  $G$  is the multiplicative group  $\mathbb{F}_p^*$ , with  $p$  a prime number, as in the Example 3.1.2, then this cryptographic protocol is not really different from the standard ElGamal cryptosystem, described in Section 1.2.2. The security is also based on the discrete logarithm problem and the Diffie-Hellman problem.

Therefore, the standard ElGamal cryptosystem is a special case of this public key cryptosystem,

Table 1.3.: Summary of **Protocol 3**: Group theoretical ElGamal like public key cryptosystem using semidirect products

<b>Public Parameters</b>	
Group $G$ and cyclic subgroup $H$ of the group $Aut(G)$ , $g \in G$ and $\phi \in H \subseteq Aut(G)$ .	
Alice	Bob
Key Creation	
Choose private key $n \in \mathbb{N}$ . Compute $(a, \phi^n) := (g, \phi)^n$ with $a := \phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g$ . Publish $a$ .	
Encryption	
	Choose plaintext $m \in G$ . Choose random ephemeral key $k \in \mathbb{N}$ . Compute $(c_1, \phi^k) := (g, \phi)^k$ with $c_1 := \phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g$ , $(a, y) \cdot (c_1, \phi^k) = \underbrace{(\phi^k(a) \cdot c_1, y \cdot \phi^k)}_{=:b}$ and $c_2 := b \cdot m = \phi^k(a) \cdot c_1 \cdot m$ . Send ciphertext $(c_1, c_2)$ to Alice. $(c_1, c_2)$
←	
Decryption	
Compute $(c_1, x) \cdot (a, \phi^n) = \underbrace{(\phi^n(c_1) \cdot a, x \cdot \phi^n)}_{=:K}$ and recover $m = K^{-1} \cdot c_2$ .	

hence, breaking this cryptosystem would imply breaking the ElGamal cryptosystem.

If the platform group is  $G = GL(r, \mathbb{K})$ , with  $\mathbb{K}$  a field, as in Example 3.1.4, it was assumed that the security is based on the discrete logarithm problem and, furthermore, the security assumption is that it is computationally hard to reclaim the “key”  $b = H^{-(n+k)}(HM)^{n+k}$  from the quadruple

$$\left( H, M, a := H^{-n}(HM)^n, c_1 := H^{-k}(HM)^k \right),$$

with  $H, M \in G$  and  $n, k \in \mathbb{N}$ . Therefore, Alice has to take care that the matrices  $H$  and  $HM$  do not commute (see Remark 3.1.5).

This example was also given in the work [Mol15], but in the time under review the paper [KMU14] by M. Kreuzer, A. D. Myasnikov and A. Ushakov appeared in which a linear algebra attack on the HKKS-key exchange protocol with platform  $G = Mat(3, \mathbb{F}_7[A_5])$  was given. Their attack affects also this cryptographic protocol. It is explained in more details in Section 3.3 and also a linear decomposition attack by V. Roman'kov is described.

**Conclusion:**

Extending a given cryptographic protocol to another cryptographic protocol is interesting, it gives more possibilities to use the suggested mathematical theory for cryptology. The research about the HKKS-scheme comprises also this scheme. A security analysis must be done for each platform group. If a platform group is found, which is not vulnerable to the known attacks, this scheme works. Therefore, the theoretical idea is interesting, but for applications a platform group must be found, which is optimal in terms of security and efficiency. D. Kahrobaei and V. Shpilrain are working on this problem, see [KS16].

The work about this cryptographic protocol leads to groups and especially to non-commutative groups, which gives input for the later newly developed cryptographic protocols which are based on combinatorial group theory.

• **Protocol 4:** Signature with a semigroup of  $3 \times 3$  matrices over  $\mathbb{F}_7[A_5]$

**Protocol 4**, the signature with  $G$  a semigroup of  $3 \times 3$  matrices over the group ring  $\mathbb{F}_7[A_5]$ , is summarized in Table 1.4 (page 28), which is exactly Table 3.4 (page 83) in Chapter 3.

Table 1.4.: Summary of **Protocol 4:** Signature with a semigroup of  $3 \times 3$  matrices over  $\mathbb{F}_7[A_5]$

<b>Public Parameters</b>	
$G$ the semigroup of $3 \times 3$ matrices with entries in $\mathbb{F}_7[A_5]$ , an invertible $H \in G$ for the automorphism $\rho_H$ and a qualified hash function $h$ .	
Alice	Bob
Choose $n \in \mathbb{N}$ and $M \in G$ privately. Compute $(a, \rho_H^n) := (M, \rho_H)^n$ with $a := \rho_H^{n-1}(M) \cdot \rho_H^{n-2}(M) \cdots \rho_H(M) \cdot M$ $\quad = H^{-n}(HM)^n$ . Take care that $a^{-1} \notin G$ and that $H$ and $HM$ do not commute.	
Public Key: $a$	
Choose message $m$ and compute value $h(m) \in G$ . Pick an ephemeral key $k$ and compute $(b, \rho_H^k) := (M, \rho_H)^k$ with $b := \rho_H^{k-1}(M) \cdot \rho_H^{k-2}(M) \cdots \rho_H(M) \cdot M$ $\quad = H^{-k}(HM)^k$ . Compute $Z := h(m) \cdot \rho_H^n(b) = h(m) \cdot H^{-n-k}(HM)^k H^n$ .	
Signature: $(k, b, Z, m)$	
	Compute $(a, x) \cdot (b, \rho_H^k) = (\underbrace{\rho_H^k(a) \cdot b, x \cdot \rho_H^k}_{=: E}),$ it is $E = H^{-(k+n)}(HM)^{n+k}$ . Prove $Z \cdot a = h(m) \cdot H^{-n-k}(HM)^{k+n}$ $\quad = h(m) \cdot E$ .

**Security:**

A detailed security analysis is given. It is based on the discrete logarithm problem. In addition it turns out, that the ephemeral key  $k$  should be used only once and should be a prime number. Alice should choose for each new signature a lesser new ephemeral key  $k$  than she uses for the previous signature. This leads to the problem, that Alice can just perform, with her private key  $n$ , a finite number of signatures, which depend on her first ephemeral key  $k_1$ .

In addition there is an ongoing research about the HKKS-scheme, which also affects **Protocol 3** and **Protocol 4**. We give an overview of this research, which comprises four research papers. It turns out, that the security assumptions, which are based on the discrete logarithm problem and Diffie-Hellman problem, especially in the case with matrices, are not sufficient for the security. V. Roman'kov in [Rom15] shows that he is able to determine the "key"  $K = H^{-n-k}(HM)^{n+k}$  with a linear decomposition attack based on the decomposition method introduced by him in monography [Rom13a] and paper [Rom13b]. He shows, that in this case, contrary to the common opinion (and some explicitly stated security assumptions), one does not need to solve the underlying algorithmic problems to break the scheme, that means, there is another algorithm that recovers the keys without solving the principal algorithmic problem (discrete logarithm problem and Diffie-Hellman problem) on which the security assumptions were first based. This changes completely the understanding of security of this cryptographic protocol. The efficacy of the attack depends on the platform group, thus it requires a specific analysis in each particular case.

**Conclusion:**

Extending the HKKS-scheme to a signature protocol gives another way to use semidirect products for cryptology. Theoretically it is interesting to get this extension, but for applications it has the disadvantage that Alice can just perform, with her private key  $n$ , a finite number of signatures, which depend on her first ephemeral key  $k_1$ . The work about this cryptographic protocol as about **Protocol 3** leads also to groups and especially to non-commutative groups, which gives input for the later newly developed cryptographic protocols which are based on combinatorial group theory.

**Further research questions for Chapter 3:**

We give some ideas for further research questions.

- Are the attacks in [Rom15] also effective against the introduced signature (**Protocol 4**)? The attacks need the element  $HM$  or  $M$  respectively, but  $M$  is a private element for Alice in the signature, therefore also  $HM$  is not known publicly. Is it possible to calculate  $Z'$ , see Security 3.2.3 (I) 2. b), with such a kind of attack?
- Find optimal platform groups in terms of security and efficiency for the ElGamal like cryptosystem under considerations of the attacks especially of the decomposition attack by V. Roman'kov.
- Find optimal platform (semi)groups in terms of security and efficiency for the HKKS-key exchange protocol under considerations of the attacks especially of the decomposition attack by V. Roman'kov.
- Find optimal platform semigroups for the signature protocol. A specific analysis is required for each platform group.
- Is it possible to develop other cryptographic protocols which use the idea behind the HKKS-key exchange protocol, for example a challenge and response system?

## Chapter 4

Chapter 4 introduces the combinatorial group theory background for the newly developed cryptographic protocols (**Protocol 6** to **Protocol 12**), which are based on this theory.

The books [CgRR08], [LS77] and [MKS66] are the basis for this chapter. The reader should be familiar with the basics of groups as it is presented in a course about algebra (see for instance [JS06]).

Combinatorial group theory is the branch of algebra which studies groups with the help of group presentations. A group presentation for a group  $G$  consists of a set  $X$  of generators and a set  $R$  of defining relators on  $X$ . We write

$$G = \langle X \mid R \rangle.$$

The group  $G$  is called finitely generated if both sets  $X$  and  $R$  are finite. The newly developed cryptographic protocols use finitely generated free groups. Let  $F$  be a finitely generated free group with free generating set  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \in \mathbb{N}$ , then the group  $F$  is the set of all reduced words in  $X^{\pm 1}$ , which is defined as  $X^{\pm 1} = \{x_1, x_1^{-1}, x_2, x_2^{-1}, \dots, x_q, x_q^{-1}\}$ , where a word is called reduced if it does not contain subwords of the form  $x_j^{-1}x_j$  or  $x_jx_j^{-1}$ ,  $1 \leq j \leq q$ . The identity is considered as the empty word, which is 1. The set of relators for a free group  $F$  consists only of trivial relators, which are of the form  $w_jw_j^{-1}$  or  $w_j^{-1}w_j$ , with  $w_j$  a word in  $X$ , thus we denote  $F$  by

$$F = \langle X \mid \ \rangle.$$

The empty space on the right symbolized, that there are only trivial relators.

### Inner structure of Chapter 4:

Chapter 4 starts with a detailed introduction of free groups and group presentations.

Among finitely generated free groups the cryptographic protocols make use of Nielsen transformations, Nielsen reduced sets and additional theory, thus these will be explained next. Nielsen transformations are a linear technique to study free groups and general infinite groups. In addition the group of all automorphisms of a free group  $F$ , denoted by  $Aut(F)$ , is generated by a regular Nielsen transformation (which is a finite product of transformation (T1) and (T2), see Definition 1.3.2) between two basis of  $F$ , and, each regular Nielsen transformation between two basis of  $F$  defines an automorphism of  $F$ .

Let  $U := \{u_1, u_2, \dots, u_t\} \subset F$ ,  $t \geq 2$ , with  $u_i$  reduced words in  $X$ .

**Definition 1.3.2.** An elementary Nielsen transformation on  $U = \{u_1, u_2, \dots, u_t\} \subset F$  is one of the following transformations

- (T1) replace some  $u_i$  by  $u_i^{-1}$ ;
- (T2) replace some  $u_i$  by  $u_iu_j$  where  $j \neq i$ ;
- (T3) delete some  $u_i$  where  $u_i = 1$ .

In all three cases the  $u_k$  for  $k \neq i$  are not changed. A (finite) product of elementary Nielsen transformations is called a **Nielsen transformation**. A Nielsen transformation is called **regular** if it is a finite product of the transformations (T1) and (T2), otherwise it is called **singular**.

**Definition 1.3.3.**

A finite set  $U$  in  $F$  is called **Nielsen reduced**, if for any three elements  $v_1, v_2, v_3$  from  $U^{\pm 1}$  the following conditions hold:

(N0)  $v_1 \neq 1$ ;

(N1)  $v_1 v_2 \neq 1$  implies  $|v_1 v_2| \geq |v_1|, |v_2|$ ;

(N2)  $v_1 v_2 \neq 1$  and  $v_2 v_3 \neq 1$  implies  $|v_1 v_2 v_3| > |v_1| - |v_2| + |v_3|$ .

Recall,  $|v|$  denotes the **free length** of  $v \in F$ , that is, the number of letters from  $X^{\pm 1}$  in the freely reduced word  $v$ . We write  $|v|_X$ , if it is not clear from which set the letters of  $v$  are and we count the letters in  $v$  which are given as elements in  $X^{\pm 1}$ .

Nielsen reduced sets of  $F$  can be seen as special basis for subgroups of  $F$ , because out of all systems of generators for this subgroup a Nielsen reduced set  $U = \{u_1, u_2, \dots, u_m\}$ ,  $u_i$  reduced words in  $X$  with special conditions, has the shortest total X-length, which is  $\sum_{i=1}^m |u_i|_X$  (with  $|u_i|_X$  the free length of  $u_i \in F$ , that is, the number of letters from  $X^{\pm 1}$  in the freely reduced word  $u_i$ ). We write  $F_U$  for a subgroup of the free group  $F = \langle X \mid \ \rangle$ , whereby the free generating set  $U$  consists of words in  $X$ , it is  $F_U = \langle U \mid \ \rangle$ .

Afterwards we explain several fundamental problems in group theory, which could be used for cryptology. For example D. Panagopoulos uses the **word problem** for his  $(n, t)$ -secret sharing scheme. The **word problem** is the following:

*Let  $G = \langle X \mid R \rangle$  be a presentation of a group and  $g \in G$  a given word in  $X$ . Determine algorithmically (in finitely many steps) if  $g$  represents the identity or not.*

Another problem is the **extended word problem**, also called **membership problem**, which is:

*Given a recursively presented group  $G$ , a subgroup  $H$  of  $G$  generated by  $h_1, h_2, \dots, h_k$  and an element  $g \in G$ , determine whether or not  $g \in H$ .*

A related problem (to the membership problem) is the **constructive membership problem**, which is:

*Given a recursively presented group  $G$ , a subgroup  $H$  of  $G$  generated by  $h_1, h_2, \dots, h_k$  and an element  $h \in H$ , find an expression of  $h$  in terms of  $h_1, h_2, \dots, h_k$ .*

The last two problems play a role for the security of the newly developed cryptographic protocols in this thesis.

The **Protocols 6-12** use automorphisms on finitely generated free groups. These automorphisms can be generated with the help of Nielsen transformations or alternatively with so called Whitehead-Automorphisms. Therefore, we close Chapter 4 introducing Whitehead-Automorphisms. With the help of these automorphisms we could develop an approach for choosing automorphisms randomly of the automorphism group  $Aut(F)$ , with  $F$  a finitely generated free group.

## Chapter 5

**Protocol 5**, which is a purely combinatorial  $(n, t)$ -secret sharing scheme, is introduced in Chapter 5. It uses the combinatorial share distribution method, which D. Panagopoulos describes in [Pan10] for his combinatorial group theoretical  $(n, t)$ -secret sharing scheme.

**Protocol 5** is published in the survey article [CFMRZ16] as research in the field of secret sharing schemes. It is also published in [MR15].

### Inner structure of Chapter 5:

We start Chapter 5 with a definition of  $(n, t)$ -secret sharing schemes and briefly explain the two first mathematical  $(n, t)$ -secret sharing schemes. One was given by A. Shamir in [Sha79] and the other by G. Blakley in [Bla79]. A. Shamir's secret sharing protocol has become the standard method for solving the  $(n, t)$ -secret sharing problem. He lists in his paper [Sha79] some useful properties for  $(n, t)$ -secret sharing schemes, which we also use to analyze different secret sharing schemes and compare them to A. Shamir's scheme.

We explain D. Panagopoulos'  $(n, t)$ -secret sharing scheme whereby we are mostly interested in the share distribution method.

D. Panagopoulos distributes (for a  $(n, t)$ -secret sharing scheme) elements of a set  $R$  which are all needed to reconstruct the secret, it is  $R = \{r_1, r_2, \dots, r_m\}$  with  $m = \binom{n}{t-1}$  and  $t \leq n$ , between  $n$  participants in the following steps:

1. Let  $n, t \in \mathbb{N}$ , with  $t \leq n$ , calculate  $m = \binom{n}{t-1}$ . Choose  $R = \{r_1, r_2, \dots, r_m\}$ , such that the secret is only reconstructible if all elements of  $R$  are known.
2. Let  $A_1, A_2, \dots, A_m$  be an enumeration of subsets of  $\{1, 2, \dots, n\}$  with  $t-1$  elements. Define  $n$  subsets  $R_1, R_2, \dots, R_n$  of the set  $\{r_1, r_2, \dots, r_m\}$  with the property

$$r_j \in R_i \iff i \notin A_j \quad \text{for } j = 1, 2, \dots, m \text{ and } i = 1, 2, \dots, n.$$

3. The participant  $p_i$  gets the share-set  $R_i$ ,  $1 \leq i \leq n$ .

If  $t$  or more participants combine their sets, they can reconstruct the set  $R$ . With the knowledge of  $R$  they are able to reconstruct the secret.

The purely combinatorial  $(n, t)$ -secret sharing protocol (**Protocol 5**) is introduced and we also give an example.

We realize that the share distribution method by D. Panagopoulos is also given as a special case by M. Ito, A. Saito and T. Nishizeki in [ISN87]. We show that if the method in [ISN87] is used to generate a  $(n, t)$ -secret sharing scheme then the same share distribution method as by D. Panagopoulos is described. M. Ito, A. Saito and T. Nishizeki use a multiple assignment scheme, which is a method to distribute to each participant more than only one share, together with a  $(m, m)$ -secret sharing scheme. Thus, we will see that the share distribution method by D. Panagopoulos is a special case of paper [ISN87].

In addition we realize that the purely combinatorial secret sharing scheme (**Protocol 5**) is very similar to a scheme, which J. Benaloh and J. Leichter obtain if they realize a  $(n, t)$ -secret sharing scheme using minimal CNF form, described in their paper [BL90].

We will explain this in detail in a section about access structures of generalized secret sharing schemes, because the papers [ISN87] and [BL90] examine such structures. Generalized secret sharing schemes realize not only the situation where arbitrary  $t$  of  $n$  persons should be able to reconstruct a secret ( $(n, t)$ -secret sharing scheme) but also some more special structures. For



example we assume that in a company with two directors and three vice-directors a secret should be reconstructed if two directors or three vice-directors or one director and two vice-directors of the company cooperate.

We close Chapter 5 comparing the CFRZ-scheme, D. Panagopoulos' scheme and the purely combinatorial  $(n, t)$ -secret sharing scheme to Shamir's scheme.

• **Protocol 5:** Purely combinatorial  $(n, t)$ -secret sharing scheme

**Protocol 5** uses a method for the distribution of the shares, which D. Panagopoulos describes, the secret  $S$  is the sum of the multiplicative inverse of elements in the natural numbers, it is

$$S = \sum_{i=1}^m \frac{1}{a_i},$$

with  $a_i \in \mathbb{N}$ . This cryptographic protocol is summarized in Table 1.5 (page 33), which is very similar to Table 5.1 (page 115) in Chapter 5.

Table 1.5.: Summary of **Protocol 5:** Purely combinatorial  $(n, t)$ -secret sharing scheme

$(n, t)$ -secret sharing scheme	
Dealer	Participants $p_1, p_2, \dots, p_n$
<p>Calculate <math>m = \binom{n}{t-1}</math>.                      Choose <math>a_1, a_2, \dots, a_m \in \mathbb{N}</math>.                      Construct sets <math>R_j \subseteq \{a_1, a_2, \dots, a_m\}</math>                      with share distribution method given by                      D. Panagopoulos;                      it is <math> R_j  = \binom{n-1}{t-1}</math> for <math>j = 1, 2, \dots, n</math>.</p> <p>Distribute shares to the participants.</p>	<div style="text-align: center;"> <math>\xrightarrow{R_1} p_1</math>  <math>\xrightarrow{R_2} p_2</math>  <math>\vdots</math>  <math>\xrightarrow{R_n} p_n</math> </div> <p><math>t</math> participants combine their shares and thus                      get the set <math>\{a_1, a_2, \dots, a_m\}</math>.                      The secret is</p> $S = \sum_{i=1}^m \frac{1}{a_i}.$

**Security:**

The security depends on the distribution method of the shares and is hence analogous to the security of D. Panagopoulos share distribution method.

If just  $t - 1$  arbitrary sets (or less) of the sets  $R_1, R_2, \dots, R_n$  are combined, there exist a  $j$ , such that the element  $a_j$  is not included in the union of these sets. If just one element  $a_j$  is absent, the

participants do not reconstruct the correct sum  $S$ , and hence cannot compute the correct secret. Each  $a_j$  is in each union of at least  $t$  subsets, thus  $t$  participants get the set  $\{a_1, a_2, \dots, a_m\}$  and are able to reconstruct the secret.

**Comparison with A. Shamir's suggested properties:**

A. Shamir lists some useful properties for  $(n, t)$ -secret sharing schemes in his paper [Sha79], which we also use to analyze different secret sharing schemes and compare them to A. Shamir's scheme.

These properties for  $(n, t)$ -secret sharing schemes are the following.

- (1) The size of each piece (which are the shares for the participants) does not exceed the size of the original data (which is the secret).
- (2) When  $t$  is kept fixed, pieces can be dynamically added or deleted (for example, when executives join or leave a company) without affecting the other pieces. (A piece is deleted only when a leaving executive makes it completely inaccessible, even to himself.)
- (3) It is easy to change the pieces (the shares for the participants) without changing the original data (which is the secret). All we need is a new polynomial  $g(x)$  with the same free term. (In Shamir's secret sharing scheme, the secret is the constant term of a polynomial  $g(x)$ .) A frequent change of this type can greatly enhance security since the pieces exposed by security breaches cannot be accumulated unless all of them are values of the same edition of the polynomial  $g(x)$ .
- (4) By using tuples of polynomial values as pieces, we can get a hierarchical scheme in which the number of pieces needed to determine the secret depends on their importance. For example, if we give the company's president three values of  $g(x)$ , each vice-president two values of  $g(x)$ , and each executive one value of  $g(x)$ , then a  $(n, 3)$ -threshold scheme enables checks to be signed either by any three executives, or by any two executives one of whom is a vice-president, or by the president alone.

In addition we choose the following fifth property.

- (5) It is easy to change the secret without changing the shares of the participants.

Table 1.6 (page 35) summarized the results of the comparison between Shamir's scheme, the CFRZ-scheme, D. Panagopoulos' scheme and **Protocol 5** concerning these properties, this is exactly Table 5.4 (page 132) in Chapter 5.

Furthermore, the comparison of the running time for the participants is given in Chapter 5.

**Shamir's scheme:** The involved polynomial interpolation has a quadratic running time, that means, if we have  $t$  supporting points we get a complexity of  $\mathcal{O}(t^2)$ .

**The CFRZ-scheme:** In order to orthonormalize  $t$  linear independent vectors in a real inner product space with dimension  $m$  we have a total running time of  $\mathcal{O}(t^2m)$ .

In the CFRZ-scheme the variable  $m$  depends on the number  $t$ , because  $m > t$  is postulated. The total running time for this scheme is longer than for Shamir's.

**Panagopoulos' scheme:** The word problem in a Coxeter group, for example, is solvable within quadratic running time, due to the fact, that Coxeter groups are automatic and automatic groups have a solvable word problem with a quadratic running time.

Table 1.6.: Summary of the comparison

Shamir's properties	Shamir's scheme	CFRZ-scheme	D. Panagopoulos' scheme	<b>Protocol 5</b>
(1)	✓	✓	–	–
(2)	✓	✓	–	–
(3)	✓	✓	✓	✓
(4)	✓	✓	✓	✓
Additional property (5)	–	✓	✓	–

**Protocol 5:** For the reconstruction of the shares the participants only add up  $m$  elements. Therefore, for the participants it is just  $\mathcal{O}(m)$ , where  $m = \binom{n}{t-1}$  is already previously calculated by the dealer, and hence  $m$  is fixed for the participants.

In the special case of a  $(t+1, t)$ -secret sharing scheme the running time depends for **Protocol 5** also only on  $t$  like in Shamir's scheme. Hence, the running time is also  $\mathcal{O}(t^2)$ , but the participants only sum up  $m$  elements, which is a very easy operation to reconstruct the secret. It is important in terms of practicability, that the dealer calculates and distributes the shares for the participants in **Protocol 5** long before the secret is needed by the participants. Hence, the dealer has enough time to execute the share distribution method and his computational cost should be of no consequence for this cryptographic protocol. Note, that the dealer has to generate  $m = \binom{n}{t-1}$  shares and uses the share distribution method given by D. Panagopoulos. The size of the share-set exceeds the size of the secret but the calculation to reconstruct the secret is very easy and fast.

### Conclusion:

In contrast to other secret sharing schemes the part for the participants is very easy, they only have to add up  $m$  elements. The (time) expensive part is the part of the dealer, who has to generate the sets  $R_i$  for the participants. In contrast to Shamir's scheme, where the part of the dealer is the easier one and the participants have to do polynomial interpolation to reconstruct the secret.

## Chapter 6

Chapter 6 introduces two secret sharing schemes, which are based on Nielsen transformations (see Chapter 4). **Protocol 6** uses in addition a free subgroup of the special linear group  $\text{SL}(2, \mathbb{Q})$  and the secret is a sum over traces of matrices in a set  $M \subset \text{SL}(2, \mathbb{Q})$ . **Protocol 7** uses in addition Nielsen reduced sets and the secret is a sum, which uses the free length of elements in a Nielsen reduced set.

We present both cryptographic protocols as  $(m, m)$ -secret sharing schemes, because it is possible to modify them to any  $(n, t)$ -secret sharing scheme if the share distribution method given by D. Panagopoulos is used and  $m$  is determined as  $m = \binom{n}{t-1}$ . Both developed cryptographic protocols are published in the survey article [CFMRZ16] as research in the field of secret sharing schemes. They are also published in [MR15]

### Inner structure of Chapter 6:

Firstly, **Protocol 6** is introduced. We give an example and compare it to Shamir's suggested properties. Secondly, **Protocol 7** is introduced and we also give an example and analyze it concerning Shamir's suggested properties.

#### • Protocol 6: Secret sharing scheme using Nielsen transformations and $\text{SL}(2, \mathbb{Q})$

**Protocol 6**, as a  $(n, t)$ -secret sharing scheme, is summarized in Table 1.7 (page 37), which is very similar to Table 6.1 (page 138) in Chapter 6.

#### Security:

The secret is only reconstructible if the whole set  $M' = \{M'_1, M'_2, \dots, M'_m\}$  is known by the participants, because  $(M_1, M_2, \dots, M_m)$  and  $(M'_1, M'_2, \dots, M'_m)$  differ to each other only in the position order and inverses, that means  $M' = \{M_1^{\delta_1}, M_2^{\delta_2}, \dots, M_m^{\delta_m}\}$  with  $\delta_i \in \{1, -1\}$ . Both sets  $U$  and  $N$  are needed for the reconstruction procedure of the secret.

If only the set  $N$  is known, then the matrices in  $\text{SL}(2, \mathbb{Q})$  are known, but nobody knows which Nielsen transformations should be applied on  $N$  to get the set  $M'$ . It is also unknown how many Nielsen transformations were used. There could be hints for Nielsen transformations, if elements in  $N$  could be written in terms of other elements in  $\text{SL}(2, \mathbb{Q})$ . Therefore, the unknown solvability of the (constructive) membership problem for (discrete) free subgroups of  $\text{SL}(2, \mathbb{Q})$ , which are not subgroups in  $\text{SL}(2, \mathbb{Z})$ , play a role for the security.

#### Comparison with A. Shamir's suggested properties:

**Protocol 6** uses the share distribution method given by D. Panagopoulos to be a  $(n, t)$ -secret sharing protocol. Therefore, this scheme fulfills the same properties of Shamir as D. Panagopoulos' scheme does. That means (3) and (4) are fulfilled and (1) and (2) are not fulfilled. Furthermore, the additional property (5) does not hold (see Chapter 5 for the properties).

A variation for **Protocol 6** is given, such that the additional property (5) is fulfilled; but the dealer has to take care that all (or almost all) matrices in the set  $M$  are in  $\text{SL}(2, \mathbb{Q})$  but not in  $\text{SL}(2, \mathbb{Z})$  then the constructive membership problem cannot be used to get information about the used Nielsen transformation, due to the fact, that there is no algorithm known to solve the (constructive) membership problem for (discrete) free subgroups (with rank greater 2) in  $\text{SL}(2, \mathbb{Q})$ , which are not subgroups in  $\text{SL}(2, \mathbb{Z})$ .

In [Ste89] an algorithm, using elementary Nielsen transformations, is presented which, given a finite set  $S$  of  $m$  words of a free group, returns a set  $S'$  of Nielsen reduced words, such that  $\langle S \rangle = \langle S' \rangle$ ; This is exactly what the participants have to do to get the elements which are needed to reconstruct the secret. The algorithm runs in  $\mathcal{O}(\ell^2 m^2)$ , where  $\ell$  is the maximum free

Table 1.7.: Summary of **Protocol 6**: Secret sharing scheme using Nielsen transformations and  $\text{SL}(2, \mathbb{Q})$ 

( $n, t$ )-secret sharing scheme	
Dealer	Participants $p_1, p_2, \dots, p_n$
<p>Calculate <math>m = \binom{n}{t-1}</math>.</p> <p>Choose abstract free generating set <math>X := \{x_1, x_2, \dots, x_m\}</math> and explicit free generating set <math>M := \{M_1, M_2, \dots, M_m\}</math> with <math>M_i \in \text{SL}(2, \mathbb{Q})</math> (all or almost all <math>M_i \notin \text{SL}(2, \mathbb{Z})</math>).</p> <p>Apply simultaneously regular Nielsen transformation (NT) on <math>X</math> and <math>M</math>:</p> $\begin{array}{ccc} (x_1, x_2, \dots, x_m) & & (M_1, M_2, \dots, M_m) \\ \downarrow \text{NT} & & \downarrow \text{NT} \\ (u_1, u_2, \dots, u_m) & & (N_1, N_2, \dots, N_m) \end{array}$ <p><math>U := \{u_1, u_2, \dots, u_m\}</math>; <math>N := \{N_1, N_2, \dots, N_m\}</math>.</p> <p>Construct sets <math>R_j \subseteq U</math> and <math>S_j \subseteq N</math> with share distribution method given by D. Panagopoulos; it is <math> R_j  =  S_j  = \binom{n-1}{t-1}</math> for <math>j = 1, 2, \dots, n</math>.</p> <p>Distribute shares to the participants.</p>	<div style="text-align: center;"> <math>(R_1, S_1) \longrightarrow p_1</math>  <math>(R_2, S_2) \longrightarrow p_2</math>  <math>\vdots</math>  <math>(R_n, S_n) \longrightarrow p_n</math> </div> <p><math>t</math> participants combine their shares and thus get the sets <math>U</math> and <math>N</math>.</p> <p>Apply simultaneously regular Nielsen transformation (NT) on <math>U</math> and <math>N</math>:</p> $\begin{array}{ccc} (u_1, u_2, \dots, u_m) & & (N_1, N_2, \dots, N_m) \\ \downarrow \text{NT} & & \downarrow \text{NT} \\ (x'_1, x'_2, \dots, x'_m) & & (M'_1, M'_2, \dots, M'_m) \end{array}$ <p>The secret is</p> $S := \sum_{j=1}^m \frac{1}{ a'_j } \in \mathbb{Q}^+, \text{ with } a'_j := \text{tr}(M'_j) \in \mathbb{Q}.$

length of a word in  $S$ . In this cryptographic protocol, the dealer fixes the number  $m$ , hence the running time depends only on the maximum free length  $\ell$  of the words in a Nielsen equivalent set. Thus, the participants have a running time of  $\mathcal{O}(\ell^2)$ .

**Conclusion:**

**Protocol 6** is the first newly developed cryptographic protocol in this thesis, which uses combinatorial group theory, especially Nielsen transformations and finitely generated free groups. It is mathematically a very interesting cryptographic protocol, which serves very good as a basis to develop other cryptographic protocols. In this thesis it is the basis for **Protocol 7** to **Protocol 12**, which are also based on combinatorial group theory.

- **Protocol 7: Secret sharing scheme using Nielsen transformations together with Nielsen reduced sets and free lengths of certain words**

**Protocol 7** is given as a  $(n, t)$ -secret sharing scheme and summarized in Table 1.8 (page 39), which is very similar to Table 6.5 (page 148) in Chapter 6.

**Security:**

By combining less than  $m$  shares the participants get a subset  $\tilde{V}$  of  $V$ , it is  $|\tilde{V}| \leq m - 1$ . If they apply Nielsen transformations on the set  $\tilde{V}$  in a Nielsen reducing manner they do not get a subset  $\tilde{U}$  of  $U$ , in general. Hence, they get no useful information to reconstruct the secret.

**Comparison with A. Shamir's suggested properties:**

**Protocol 7** uses the share distribution method given by D. Panagopoulos to be a  $(n, t)$ -secret sharing protocol. Therefore, this scheme fulfills the same properties of Shamir as D. Panagopoulos' scheme does. That means (3) and (4) are fulfilled and (1) and (2) are not fulfilled. Furthermore, the additional property (5) does not hold (see Chapter 5 for the properties).

In [Ste89] an algorithm, using elementary Nielsen transformations, is presented which, given a finite set  $S$  of  $m$  words of a free group, returns a set  $S'$  of Nielsen reduced words, such that  $\langle S \rangle = \langle S' \rangle$ ; This is exactly what the participants have to do to get the elements which are needed to reconstruct the secret. The algorithm runs in  $\mathcal{O}(\ell^2 m^2)$ , where  $\ell$  is the maximum free length of a word in  $S$ . In this cryptographic protocol, the dealer fixes the number  $m$ , hence the running time depends only on the maximum free length  $\ell$  of the words in a Nielsen equivalent set. Thus, the participants have a running time of  $\mathcal{O}(\ell^2)$ .

**Conclusion:**

**Protocol 7** is, like **Protocol 6**, mathematically a very interesting cryptographic protocol which in addition uses a Nielsen reduced subset  $U \neq X$  of a finitely generated free group  $F = \langle X \mid \ \ \rangle$  and gives therefore the final input for the newly developed cryptosystems.

Table 1.8.: Summary of **Protocol 7**: Secret sharing scheme using Nielsen transformations together with Nielsen reduced sets and free lengths of certain words

( $n, t$ )-secret sharing scheme	
Dealer	Participants $p_1, p_2, \dots, p_n$
<p>Calculate <math>m = \binom{n}{t-1}</math>.            Choose abstract free generating set <math>X = \{x_1, x_2, \dots, x_q\}</math>, <math>q \in \mathbb{N} \setminus \{1\}</math>, and a Nielsen reduced set <math>U = \{u_1, u_2, \dots, u_m\} \subset F</math>, <math>u_i</math> words in <math>X</math>.</p> <p>Apply regular Nielsen transformation (NT) on <math>U</math>:</p> $\begin{array}{c} (u_1, u_2, \dots, u_m) \\ \downarrow \text{NT} \\ (v_1, v_2, \dots, v_m) \end{array}$ <p><math>V := \{v_1, v_2, \dots, v_m\}</math>.</p> <p>Construct sets <math>R_j \subseteq V</math> with share distribution method given by D. Panagopoulos; it is <math> R_j  = \binom{n-1}{t-1}</math> for <math>j = 1, 2, \dots, n</math>.</p> <p>Distribute shares to the participants.</p>	<div style="text-align: center;"> <math display="block">\begin{array}{ccc} \xrightarrow{R_1} &amp; &amp; p_1 \\ \xrightarrow{R_2} &amp; &amp; p_2 \\ &amp; \vdots &amp; \\ \xrightarrow{R_n} &amp; &amp; p_n \end{array}</math> </div> <p><math>t</math> participants combine their shares and thus get the set <math>V</math>.</p> <p>Apply regular Nielsen transformation (NT) on <math>V</math> to get a Nielsen reduced set:</p> $\begin{array}{c} (v_1, v_2, \dots, v_m) \\ \downarrow \text{NT} \\ (u'_1, u'_2, \dots, u'_m) \end{array}$ <p>The secret is</p> $S = \sum_{i=1}^m \frac{1}{ u'_i _X}.$

## Chapter 7

Chapter 7 introduces **Protocol 8**, which is a private key cryptosystem, and three modifications of it. This cryptographic protocol is based on combinatorial group theory, see Chapter 4, and uses a finitely generated free group  $F$ , a subgroup  $F_U$  of  $F$  with finite rank, a Nielsen reduced set and automorphisms of  $F$ .

The automorphisms are out of a common set  $\mathcal{F}_{Aut} \subset Aut(F)$ . Assume Alice sends a message to Bob. For decryption Bob needs to know which automorphisms of  $\mathcal{F}_{Aut}$  were used for the encryption procedure by Alice. For this choice of elements in  $\mathcal{F}_{Aut}$  regulations are needed. Therefore, Alice and Bob make use of a linear congruence generator with maximal periodic length.

**Protocol 8** is published in [MR16].

### Inner structure of Chapter 7:

First, we shortly introduce a linear congruence generator. **Protocol 8** is explained next. There are two possibilities to decrypt a ciphertext. We give an example for each possibility. Afterwards three modifications for this cryptographic protocol are given.

In **Protocol 8** the ciphertext is a sequence of reduced words in  $X$  where the end of each ciphertext unit is marked (for example with “ $\wr$ ”) and  $X$  is a free generating set for a free group  $F$  of finite rank. The first modification is given, in which the ciphertext is now only one reduced word in  $X$  instead of a sequence of words, in this case it is possible that additional information is needed for decryption, thus these is sent with the ciphertext if required. In the second modification a faithful representation from  $F$  into the special linear group  $SL(2, \mathbb{Q})$  is used, such that the ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$ . The third modification utilizes the negative solution of Hilbert’s Tenth Problem (see [Hil02]). Instead of a presentation of the ciphertext as a sequence of matrices in  $SL(2, \mathbb{Q})$  the ciphertext is represented as a sequence of matrices in  $GL(2, R)$  with  $R := \mathbb{Z}[y_1, y_2, \dots, y_n]$ , the integral polynomial ring in  $n \geq 2$  variables.

We close Chapter 7 with a detailed look at chosen plaintext and chosen ciphertext attacks.

### • Protocol 8: Private key cryptosystem with $Aut(F)$

There are two ways to decrypt a ciphertext given by **Protocol 8**. One uses inverses of automorphisms, which were used for encryption, and the other uses a table like Table 1.9 (page 40), which is exactly Table 7.1 (page 157) in Chapter 7.

Given the knowledge of a set  $U = \{u_1, u_2, \dots, u_N\}$ , which is part of the private parameters, the linear congruence generator  $h$  and the number  $z$ , the receiver is able to compute for each automorphism  $f_{x_i}$ ,  $i = 1, 2, \dots, z$ , the set  $U_{f_{x_i}} = \{f_{x_i}(u_1), f_{x_i}(u_2), \dots, f_{x_i}(u_N)\}$ . This is used for Table 1.9 (page 40), which is exactly Table 7.1 (page 157) in Chapter 7.

Table 1.9.: Table for decryption: Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  corresponding to ciphertext alphabet  $U_{f_{x_i}}$  depending on the automorphisms  $f_{x_i}$

	$U_{f_{x_1}}$	$U_{f_{x_2}}$	$\dots$	$U_{f_{x_z}}$
$a_1$	$f_{x_1}(u_1)$	$f_{x_2}(u_1)$	$\dots$	$f_{x_z}(u_1)$
$a_2$	$f_{x_1}(u_2)$	$f_{x_2}(u_2)$	$\dots$	$f_{x_z}(u_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_N$	$f_{x_1}(u_N)$	$f_{x_2}(u_N)$	$\dots$	$f_{x_z}(u_N)$



**Protocol 8** is summarized in Table 1.10 (page 41), which is very similar to Table 7.2 (page 158) in Chapter 7.

Table 1.10.: Summary of **Protocol 8**: Private key cryptosystem with  $Aut(F)$

<b>Public Knowledge</b>	
$F = \langle X \mid \ \rangle$ , $X = \{x_1, x_2, \dots, x_q\}$ , $q \geq 2$ ; plaintext alphabet $A = \{a_1, a_2, \dots, a_N\}$ , $N \geq 2$ ; set $\mathcal{F}_{Aut} \subset Aut(F)$ ; linear congruence generator $h$ of maximal periodic length.	
<b>Alice</b>	<b>Bob</b>
Private keys	
Nielsen reduced set $U \subset F$ , $ U  = N$ ; seed $f_{\bar{\alpha}} \in \mathcal{F}_{Aut}$ , one-to-one correspondence $A \rightarrow U$ , $a_j \mapsto u_j$ .	
Encryption	
<p>Choose message</p> $S = s_1 s_2 \cdots s_z, \quad z \geq 1,$ <p>with <math>s_i \in A</math>.</p> <p>Calculate</p> $x_1 = \bar{\alpha}, x_2 = h(x_1), \dots, x_z = h(x_{z-1})$ , obtain $f_{x_1}, f_{x_2}, \dots, f_{x_z}$ . <p>Encryption procedure:</p> <p>if <math>s_i = a_t</math> then <math>s_i \mapsto c_i := f_{x_i}(u_t)</math>, <math>1 \leq i \leq z</math>, <math>1 \leq t \leq N</math>.</p> <p>Ciphertext:</p> $C = f_{x_1}(s_1) f_{x_2}(s_2) \cdots f_{x_z}(s_z) = c_1 c_2 \cdots c_z.$	$\xrightarrow{C=c_1 c_2 \cdots c_z}$
Decryption	
	<p>Compute <math>z</math> automorphisms:</p> $x_1 = \bar{\alpha}, x_2 = h(x_1), \dots, x_z = h(x_{z-1})$ , obtain $f_{x_1}, f_{x_2}, \dots, f_{x_z}$ . <p>Two possibilities:</p> <ol style="list-style-type: none"> <li>1. For each <math>f_{x_i}</math>, <math>i = 1, 2, \dots, z</math>, compute the inverse automorphism <math>f_{x_i}^{-1}</math>.</li> <li>2. For each <math>f_{x_i}</math>, <math>i = 1, 2, \dots, z</math>, compute <math>U_{f_{x_i}} = \{f_{x_i}(u_1), f_{x_i}(u_2), \dots, f_{x_i}(u_N)\}</math> and get a table like Table 1.9 (page 40). (Decryption: Search in this table.)</li> </ol> <p>With knowledge of Table 1.9 (page 40) or inverse automorphisms <math>f_{x_i}^{-1}</math>, respectively, the decryption is as follows:</p> <p>if <math>c_i = f_{x_i}(u_t)</math> then <math>c_i \mapsto s_i := f_{x_i}^{-1}(c_i) = a_t</math>, <math>1 \leq i \leq z</math>, <math>1 \leq t \leq N</math>.</p> <p>Plaintext message</p> $S = f_{x_1}^{-1}(c_1) f_{x_2}^{-1}(c_2) \cdots f_{x_z}^{-1}(c_z)$ $= s_1 s_2 \cdots s_z, \text{ with } s_i \in A.$

Examples for **Protocol 8** are given:

1. An example is given, in which for decryption the inverse automorphisms of the  $z$  automorphisms  $f_{x_i}$ , which Alice used for encryption, are calculated, see the example attached in Appendix C.7.
2. An example is given, in which for decryption a table (see Table 1.9 (page 40)) is used, which stores the ciphertext alphabet  $U_{f_{x_i}}$  and is generated with the automorphisms Alice used for encryption, see the Example 7.0.7.

### Security:

The cryptosystem is a polyalphabetic system, that is, a word  $u_i \in U$ , and hence a letter  $a_i \in A$ , is encrypted differently at different positions in the plaintext, because different automorphisms are used during the encryption procedure for each ciphertext unit. Thus, for the ciphertext, a statistical frequency attack (see for instance [BFKR15]) over the frequency of words, which corresponds to letters in the plaintext alphabet, or groups of words, is useless.

The security depends on the fact, that the set  $U$  is private. Note, that in general  $c_i \notin F_U$ , with  $F_U = \langle U \mid \rangle$ . An eavesdropper, Eve, assumes that the elements of the set  $U$ , which were used for the encryption, can be found in the ball  $B(F, L)$  of the Cayley graph from  $F$ , with  $L = \sum_{i=1}^z |c_i|$  and  $c_i$  ciphertext units of an intercepted ciphertext

$$C = c_1 \wr c_2 \wr \cdots \wr c_z.$$

The symbol “ $\wr$ ” marks the end of each ciphertext unit  $c_i$ ,  $1 \leq i \leq z - 1$ .

The ball  $B(F, L)$  contains all elements of  $F$  with a free length equal to or less than  $L$ . The number of elements which are candidates for the set  $U$ , so called primitive elements, grows exponentially with the free length, here with  $L$ . Eve has to test subsets  $V_i$  of  $K \geq N$  elements of the ball  $B(F, L)$  to get candidates for  $U$ . For this she constructs the corresponding Nielsen reduced sets  $V'_i$  to  $V_i$  (which are minimal concerning a lexicographical order). If  $|V'_i| = N$  then  $V'_i$  is a candidate for  $U$ .

The running time is within  $\mathcal{O}(\lambda^2 K^2)$ , with  $\lambda := \max\{|v_{i_\ell}|_X \mid v_{i_\ell} \in V_i \text{ for } \ell = 1, 2, \dots, K\} \leq L$ , to get a Nielsen reduced set  $V'_i$  from  $V_i$  with a known algorithm.

Furthermore, the security depends on the way how Alice and Bob choose the automorphisms of the set  $\mathcal{F}_{Aut}$ . To verify, whether a candidate set  $V'_i$  is very likely the set  $U$  used by Alice and Bob, it is likely that Eve tests the automorphisms in  $\mathcal{F}_{Aut}$  with her set  $V'_i$  to get the ciphertext. The set  $\mathcal{F}_{Aut}$  should be large enough to make this kind of brute force search inefficient. A variation is given where the set  $\mathcal{F}_{Aut}$  is partly or fully publicly unknown, to avoid such a search. This also avoids brute force attacks which could be done over the inverse automorphisms of the set  $\mathcal{F}_{Aut}$  and known ciphertexts.

Thus, the main security certification depends on the fact, that for a single subset of  $K \geq N$  elements Eve finds a Nielsen reduced set in the running time  $\mathcal{O}(\lambda^2 K^2)$  but she has to test all possible subsets of  $K$  elements for which she needs exponential running time.

### Modification:

To improve the security certification we give three modifications.

1. We present a modification (Section 7.1) where the ciphertext is only one reduced word in  $X$  instead of a sequence of words, in this case it is possible that additional information is needed for decryption, thus these is sent with the ciphertext if required.

**Security:** The security certification is extended to the fact, that Eve is in general not able to identify the beginning and end of a ciphertext unit  $c_i$ ,  $i = 1, 2, \dots, z$ . There could also be cancellations, which she is not able to recognize. Thus, the attack which uses the

automorphisms of  $\mathcal{F}_{Aut}$  in the unmodified cryptographic protocol is not realizable in this modification.

2. We present a modification (Section 7.2), which uses a faithful representation from  $F$  into the special linear group  $SL(2, \mathbb{Q})$ , such that the ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$ . A variation (Variation 7.2.3) is given where the ciphertext is not a sequence of matrices but a sequence of entries of matrices. This reduces the space for the ciphertext and the memory space for the decryption table, like Table 1.9 (page 40).

**Security:** The security certification is extended to the fact, that there is no algorithm known to solve the (constructive) membership problem for (discrete) free subgroups of  $SL(2, \mathbb{Q})$ , which are of rank greater than or equal to 2 and not subgroups of  $SL(2, \mathbb{Z})$ . Therefore, the attack which uses a Cayley graph and automorphisms of  $\mathcal{F}_{Aut}$  in the unmodified cryptographic protocol is not realizable in this modification.

3. We present a modification (Section 7.3), which utilizes the negative solution of Hilbert's Tenth Problem. Instead of a presentation of the ciphertext as a sequence of matrices in  $SL(2, \mathbb{Q})$  the ciphertext is represented as a sequence of matrices in  $GL(2, R)$ , with  $R := \mathbb{Z}[y_1, y_2, \dots, y_n]$ , the integral polynomial ring in  $n \geq 2$  variables.

**Security:** The security certification is extended to Hilbert's Tenth Problem. In addition the security is improved by the fact, that for each encryption Alice and Bob can take privately ephemeral matrices in  $GL(2, R)$ ,  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ , with the property that the common private point  $D \in \mathbb{Z}^n$  generates the correct matrices in  $PSL(2, \mathbb{Z})$ . This gives randomness to ciphertexts, which complicates attacks for Eve. The attack which uses a Cayley graph and automorphisms of  $\mathcal{F}_{Aut}$  in the unmodified cryptographic protocol is not realizable in this modification.

#### Conclusion concerning chosen plaintext attacks and chosen ciphertext attacks:

We also analyze this private key cryptosystem as well as the modifications concerning chosen plaintext attacks (Section 7.4) and chosen ciphertext attacks (Section 7.5). If the ciphertext is given as a matrix, the system is secure against chosen plaintext attacks and chosen ciphertext attacks. If the ciphertext is a word in  $X$  it could be possible that an eavesdropper can get hints for the elements in  $U$  and hence the search for the primitive elements in the Cayley graph as well as the search for the automorphisms in  $\mathcal{F}_{Aut}$  could be performed in a more selective measure.

#### Conclusion:

Especially, the modifications of **Protocol 8** with matrices are of interest for group based cryptography. If **Protocol 8** is used together with the second modification, which uses a faithful representation into  $SL(2, \mathbb{Q})$ , then the system is secure and the security depends on the unknown solution of the (constructive) membership problem in the used matrix groups. If **Protocol 8** is used together with the third modification, which uses matrices in  $GL(2, R)$ ,  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  with  $n \geq 2$ , then the system is secure and the security depends additionally on the negative solution of Hilbert's Tenth Problem (see [Hil02]). Moreover, we also get randomness to each ciphertext by the ephemeral matrices, which the encrypter used for encryption. To generate these ephemeral matrices he only needs the common secret point  $D \in \mathbb{Z}^n$ , this improves also the security. Altogether, we get interesting new private key cryptosystems, which use non-commutative groups and are based on combinatorial group theory and not only on number theory.

### Chapter 8

Chapter 8 introduces **Protocol 9**, which is a private key cryptosystem similar to **Protocol 8**. We also give three modifications, which uses the same ideas for the modifications of **Protocol 8**. Like **Protocol 8** it is based on combinatorial group theory (see Chapter 4). It uses a

finitely generated free group  $F$ , a subgroup  $F_U$  of  $F$  with finite rank, a Nielsen reduced set and automorphisms of  $F_U$ . It differs to **Protocol 8** only in the way that it uses automorphisms of a finitely generated subgroup  $F_U$  of  $F$  instead of automorphisms of the finitely generated free group  $F$ . The automorphisms are out of a common set  $\mathcal{H}_{Aut} \subset Aut(H)$ , with an abstract free generating set  $H$  and the cardinality  $|H| = |A| = N$ , with  $A$  the set of plaintext letters. Assume Alice sends a message to Bob. For decryption Bob needs to know which automorphisms of  $\mathcal{H}_{Aut}$  were used for the encryption procedure by Alice. For this choice of elements in  $\mathcal{H}_{Aut}$  regulations are needed. Therefore, Alice and Bob make use of a linear congruence generator with maximal periodic length as for **Protocol 8**. Hence, this is explained in Chapter 7.

**Inner structure of Chapter 8:**

First **Protocol 9** is introduced. There are two possibilities to decrypt a ciphertext. We give an example for each possibility. Afterwards three modifications for this cryptographic protocol are given. We close Chapter 8 with a detailed look at chosen plaintext and chosen ciphertext attacks.

• **Protocol 9:** Private key cryptosystem with  $Aut(F_U)$

Let  $F$  be a finitely generated free group with free generating set  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ , and  $F_U$  the subgroup of  $F$  freely generated by the Nielsen reduced set  $U = \{u_1, u_2, \dots, u_N\}$ , with  $N \geq 2$  and  $u_i$  words in  $X$ . The ciphertext  $C$  can be interpreted as a sequence of words in  $X$  or of words in  $U$ . It is sent as a sequence of words in  $X$ . There are two ways to decrypt a ciphertext. One uses a table like Table 1.11 (page 44), which is exactly Table 8.1 (page 190) in Chapter 8.

Being aware of the set  $U = \{u_1, u_2, \dots, u_N\}$ , the linear congruence generator  $h$  and the number  $z$ , the decrypter is able to compute for each automorphism  $f_{u_i}$ ,  $i = 1, 2, \dots, z$ , the set

$$U_{f_{u_i}} = \{f_{u_i}(u_1), f_{u_i}(u_2), \dots, f_{u_i}(u_N)\},$$

with  $f_{u_i}(u_j)$  written as a reduced word in  $X$ . This is used for Table 1.11 (page 44).

Table 1.11.: Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  corresponding to ciphertext alphabet  $U_{f_{u_i}}$  depending on the automorphisms  $f_{u_i}$

	$U_{f_{u_1}}$	$U_{f_{u_2}}$	$\dots$	$U_{f_{u_z}}$
$a_1$	$f_{u_1}(u_1)$	$f_{u_2}(u_1)$	$\dots$	$f_{u_z}(u_1)$
$a_2$	$f_{u_1}(u_2)$	$f_{u_2}(u_2)$	$\dots$	$f_{u_z}(u_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_N$	$f_{u_1}(u_N)$	$f_{u_2}(u_N)$	$\dots$	$f_{u_z}(u_N)$

The other way, to decrypt a ciphertext, uses the Nielsen reduced set  $U$  and an algorithm to write the ciphertext units  $c_i$  (given as words in  $X$ ) as words in  $U$ . Now, together with the used automorphisms, the ciphertext can be decrypted correctly.

**Protocol 9** is summarized in Table 1.12 (page 45), which is very similar to Table 8.2 (page 191) in Chapter 8.

Table 1.12.: Summary of **Protocol 9**: Private key cryptosystem with  $Aut(F_U)$

<b>Public Knowledge</b>	
$F = \langle X \mid \ \rangle$ , $X = \{x_1, x_2, \dots, x_q\}$ , $q \geq 2$ ; plaintext alphabet $A = \{a_1, a_2, \dots, a_N\}$ , $N \geq 2$ ; abstract free group $H = \langle U \mid \ \rangle$ , $U = \{u_1, u_2, \dots, u_N\}$ with $u_i$ abstract letters; set $\mathcal{H}_{Aut} \subset Aut(H)$ ; linear congruence generator $h$ of maximal periodic length.	
<b>Alice</b>	<b>Bob</b>
Private keys	
Explicit set $U = \{u_1, u_2, \dots, u_N\}$ with $u_i$ words in $X$ , $U \subset F$ Nielsen reduced set, $ U  = N$ ; seed $f_{\bar{a}} \in \mathcal{F}_{Aut}$ , one-to-one correspondence $A \rightarrow U$ , $a_j \mapsto u_j$ .	
Encryption	
Choose message  $S = s_1 s_2 \cdots s_z, \quad z \geq 1,$ with $s_i \in A$ . Calculate $u_1 = \bar{a}, u_2 = h(u_1), \dots, u_z = h(u_{z-1})$ , obtain $f_{u_1}, f_{u_2}, \dots, f_{u_z}$ . Encryption procedure: if $s_i = a_t$ then $s_i \mapsto c_i := f_{u_i}(u_t)$ , $1 \leq i \leq z$ , $1 \leq t \leq N$ . Ciphertext: $C = f_{u_1}(s_1) f_{u_2}(s_2) \cdots f_{u_z}(s_z) = c_1 c_2 \cdots c_z$ , with $c_i$ written as words in $X$ .  $\xrightarrow{C=c_1 c_2 \cdots c_z}$	
Decryption	
	Compute $z$ automorphisms: $u_1 = \bar{a}, u_2 = h(u_1), \dots, u_z = h(u_{z-1})$ , obtain $f_{u_1}, f_{u_2}, \dots, f_{u_z}$ . Two possibilities: 1. For each $f_{u_i}$ , $i = 1, 2, \dots, z$ , compute $U_{f_{u_i}} = \{f_{u_i}(u_1), f_{u_i}(u_2), \dots, f_{u_i}(u_N)\}$ and get a table like Table 1.11 (page 44). (Decryption: Search in this table.) If $c_i = f_{u_i}(u_t)$ then $c_i \mapsto s_i = a_t$ , $1 \leq i \leq z$ , $1 \leq t \leq N$ . 2. Use Nielsen reduced set $U$ and an algorithm to write the ciphertext units $c_i$ (given as words in $X$ ) as words in $U$ . Together with the used automorphisms the ciphertext is decrypted correctly.  Reconstruct plaintext message $S = s_1 s_2 \cdots s_z$ , with $s_i \in A$ .

Examples for **Protocol 9** are given:

1. An example is given, in which for decryption a table (see Table 1.11 (page 44)) is used, which stores the ciphertext alphabet  $U_{f_{u_i}}$  and is generated with the automorphisms Alice uses for encryption, see Example 8.0.4.
2. An example is given, in which Bob knows the Nielsen reduced set  $U$ , hence with a known algorithm he is able to write the ciphertext as a sequence of words in  $U$ . With the automorphisms, Alice used for encryption, he is able to decrypt the ciphertext correctly, see the example attached in Appendix C.9.

### Security:

The cryptosystem is a polyalphabetic system, that is, a word  $u_i \in U$ , and hence a letter  $a_i \in A$ , is encrypted differently at different positions in the plaintext, because different automorphisms are used during the encryption procedure for each ciphertext unit. Thus, for the ciphertext, a statistical frequency attack (see for instance [BFKR15]) over the frequency of words, which corresponds to letters in the plaintext alphabet, or groups of words, is useless.

The security depends on the fact, that the set  $U$  is private. Note, that the ciphertext units  $c_i$  are elements in  $F_U$ , with  $F_U = \langle U \mid \rangle$ . An eavesdropper, Eve, knows that the elements of the set  $U$ , which were used for the encryption, can be found in the ball  $B(F, L_1)$  of the Cayley graph from  $F$ , with  $L_1 = \max\{|c_i|_X \mid i = 1, 2, \dots, z\}$  and  $c_i$  ciphertext units of an intercepted ciphertext

$$C = c_1 \wr c_2 \wr \dots \wr c_z.$$

The symbol “ $\wr$ ” marks the end of each ciphertext unit  $c_i$ ,  $1 \leq i \leq z - 1$ .

Let

$$\tilde{C} = \{c_1, c_2, \dots, c_z\}$$

be the set of ciphertext units and let  $\tilde{C}_{Nred}$  be a Nielsen reduced set of  $\tilde{C}$ , hence the group  $F_{\tilde{C}_{Nred}}$ , generated by  $\tilde{C}_{Nred}$ , is a free subgroup of  $F_U$  and  $rank(F_{\tilde{C}_{Nred}}) \leq z$ . The main security certification depends on the fact, that for a single subset  $V$  of  $F_U$  with  $K$  elements Eve finds a Nielsen reduced set in the running time  $\mathcal{O}(\lambda^2 K^2)$ , with  $\lambda$  the maximum over the free length of the elements in the subset  $V$  with  $K$  primitive elements, but she has to test all possible subsets of  $K$  elements for which she needs exponential running time, because the number of primitive elements grows exponentially with the free length, here with  $L_1$ . She searches in a ball  $B(F, L_1)$ , with  $L_1 = \max\{|c_i| \mid c_i \in \tilde{C}\}$  for these primitive elements.

A subset of  $V$  is also known, it is  $\tilde{C}_{Nred} \subset V$  but she has to put all other primitive elements to this set and proves if  $V'$ , which is Nielsen reduced to  $V$ , is of order  $N$  and hence a candidate for  $U$ .

Furthermore, the security depends on the way how Alice and Bob choose the automorphisms of the set  $\mathcal{H}_{Aut}$ . To verify, whether a candidate set  $V'$  is very likely the set  $U$  used by Alice and Bob, it is likely that Eve writes the ciphertext units  $c_i$  with letters of her candidate set  $V'^{\pm 1}$ . This is possible because the constructive membership problem is solvable in abstract free groups and Nielsen reduced sets. Thus, she could get hints for the automorphisms used for encryption and it is not only a brute force search through the set  $\mathcal{H}_{Aut}$ .

### Modifications:

The modifications use the ideas behind the modifications of **Protocol 8**.

1. We present a modification (Section 8.1) where the ciphertext is only one reduced word in  $X$  instead of a sequence of words, in this case it is possible that additional information is needed for decryption, thus this is sent with the ciphertext if required. The ciphertext can be interpreted as words in  $X$  and as words in  $U$ , thus the additional information could

be given about the ciphertext written as a word in  $U$  or as a word in  $X$ .

**Security:** The security certification is extended to the fact that Eve is in general not able to identify the beginning and end of a ciphertext unit  $c_i$ ,  $i = 1, 2, \dots, z$ . There could also be cancellations, which she is not able to recognize. Eve is neither able to determine the number  $L_1$  because she does not know what the ciphertext units  $c_i$  exactly look like, nor is she able to generate the set  $\tilde{C}_{Nred}$ . This worsens her attacks of the unmodified cryptographic protocol above.

2. We present a modification (Section 8.2), which uses a faithful representation from  $F$  into the special linear group  $SL(2, \mathbb{Q})$ , such that the ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$ . Furthermore, a variation (like Variation 7.2.3) can be used, where the ciphertext is not a sequence of matrices but a sequence of entries of matrices. This reduces the space for the ciphertext and the memory space for the decryption table.

**Security:** The security certification is extended to the fact, that there is no algorithm known to solve the (constructive) membership problem for (discrete) free subgroups of  $SL(2, \mathbb{Q})$ , which are of rank greater than or equal to 2 and not subgroups of  $SL(2, \mathbb{Z})$ . Therefore, the attack which uses a Cayley graph and automorphisms of  $\mathcal{F}_{Aut}$  in the unmodified cryptographic protocol is not realizable in this modification.

3. We present a modification (Section 8.3), which utilizes the negative solution of Hilbert's Tenth Problem. Instead of a presentation of the ciphertext as a sequence of matrices in  $SL(2, \mathbb{Q})$  the ciphertext is represented as a sequence of matrices in  $GL(2, R)$ , with  $R := \mathbb{Z}[y_1, y_2, \dots, y_n]$ , the integral polynomial ring in  $n \geq 2$  variables. Here we get two subcases, the first applies the modification with Hilbert's Tenth Problem on a text given as a sequence of words in  $X$  and the second applies it to a text given as a sequence of words in  $U$ .

**Security:** The security certification is extended to Hilbert's Tenth Problem. In addition the security is improved by the fact, that for each encryption Alice and Bob can take privately ephemeral matrices in  $GL(2, R)$ ,  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ , with the property that the common private point  $D \in \mathbb{Z}^n$  generates the correct matrices in  $PSL(2, \mathbb{Z})$ . This gives randomness to ciphertexts, which complicates attacks for Eve. The attack which uses a Cayley graph and automorphisms of  $\mathcal{F}_{Aut}$  in the unmodified cryptographic protocol is not realizable in this modification.

#### **Conclusion concerning chosen plaintext attacks and chosen ciphertext attacks:**

If the ciphertext is given as a matrix, the system is secure against chosen plaintext attacks (Section 8.4). If the ciphertext is a word in  $X$  it could be possible that an eavesdropper can get hints for the elements in  $U$  and hence the search for the primitive elements in the Cayley graph could be performed in a more selective measure, but these hints can also be seen in a ciphertext only attack. Hence, this is no information which only appears at a chosen plaintext attack. This cryptosystem is secure against chosen ciphertext attacks (Section 8.5). An attacker gets no additional hints for the set  $U$  than he gets with a ciphertext only attack.

#### **Conclusion:**

Due to the fact, that **Protocol 9** differs to **Protocol 8** only in the way, that it uses automorphisms of a finitely generated subgroup  $F_U$  of  $F$  instead of automorphisms of the finitely generated free group  $F$ , we get a similar cryptographic protocol. The modifications use the same ideas as the modifications for **Protocol 8**. Therefore, as above, especially the modifications of **Protocol 9** with matrices are of interest for cryptography. If **Protocol 9** is used together with the second modification, which uses a faithful representation into  $SL(2, \mathbb{Q})$ , then the system is secure and the security depends on the unknown solution of the (constructive) membership

problem in the used matrix groups. If **Protocol 9** is used together with the third modification, which uses matrices in  $\text{GL}(2, R)$ ,  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ ,  $n \geq 2$ , then the system is secure and the security depends in addition on the negative solution of Hilbert's Tenth Problem. Moreover, we get also randomness to each ciphertext by the ephemeral matrices, which the encrypter used for encryption. To generate these ephemeral matrices he only needs the common secret point  $D \in \mathbb{Z}^n$ , this improves also the security. Altogether, we get interesting new private key cryptosystems, which use non-commutative groups and are based on combinatorial group theory and not only on number theory.

## Chapter 9

Chapter 9 introduces **Protocol 10**, a symmetric key cryptosystem. It is based on combinatorial group theory, uses automorphisms of finitely generated free groups, Nielsen reduced sets and a faithful representation of a finitely generated free group  $F$  into  $\text{SL}(2, \mathbb{Q})$ . The automorphisms are out of a common set  $\mathcal{G}_{Aut} \subset \text{Aut}(G)$  (with  $G$  an abstract free group of finite rank). For decryption Bob needs to know which automorphisms of  $\mathcal{G}_{Aut}$  were used for the encryption procedure by Alice. Therefore, Alice and Bob make use of a linear congruence generator with maximal periodic length as for **Protocol 8** and **Protocol 9**. Hence, for linear congruence generators see Chapter 7.

The main difference of the cryptographic protocol in this chapter to **Protocol 8** and **Protocol 9** is, that it uses automorphisms of  $\text{Aut}(G)$  on plaintext sequences instead of automorphisms on  $F$  or  $F_U$ , respectively. Moreover, **Protocol 10** contains special random matrices, which generate randomness for the ciphertext and work as ephemeral keys in the encryption procedure.

### **Inner structure of Chapter 9:**

We first describe **Protocol 10** in a restricted version to explain the idea. This is then generalized. A variation and an example are given. The chapter closes with a closer look at chosen ciphertext and chosen plaintext attacks.

#### • **Protocol 10:** Private key cryptosystem using automorphisms on plaintext sequences

To describe this private key cryptosystem we assume that  $z = 4 \cdot t$ , with  $t \in \mathbb{N}$ , and that the letters in each plaintext part  $S_i$  are pairwise different.

**Protocol 10** is summarized in Table 1.13 (page 49) and Table 1.14 (page 50), which are exactly Table 9.1 (page 219) and Table 9.2 (page 220) in Chapter 9.



Table 1.13.: Summary of **Protocol 10**: Private key cryptosystem using automorphisms on plaintext sequences I

<b>Public Knowledge</b>	
Abstract free group $G = \langle Y \mid \ \rangle$ , $Y = \{y_1, y_2, \dots, y_5\}$ ; plaintext alphabet $A = \{a_1, a_2, \dots, a_N\}$ with $N \geq 2$ ; subset $\mathcal{G}_{Aut} \subset Aut(G)$ ; linear congruence generator $h$ of maximal periodic length.	
<b>Alice</b>	<b>Bob</b>
Private keys	
Free group $F = \langle X \mid \ \rangle$ , $X = \{x_1, x_2, \dots, x_q\}$ , $q \geq 2$ , free subgroup $F_U = \langle U \mid \ \rangle$ of $F$ with Nielsen reduced set $U = \{u_1, u_2, \dots, u_{2N}\} \subset F$ ; faithful representation $\varphi : F \rightarrow SL(2, \mathbb{Q})$ ; $F_{U'} = \langle U' \mid \ \rangle$ , $U' = \{V_1, V_2, \dots, V_{2N}\}$ with $V_i = \varphi(u_i)$ ; assignment $a_i \hat{=} V_j \iff j \equiv i \pmod{N}$ and starting seed $g_{\bar{\alpha}} \in \mathcal{G}_{Aut}$ .	
Encryption	
Choose message $S = s_1 s_2 \dots s_z$ , $z \geq 1$ and $s_i \in A$ .  Cut message into parts of $rank(G) - 1 = 4$ letters $S = \underbrace{s_1 s_2 s_3 s_4}_{S_1} \mid \underbrace{s_5 s_6 s_7 s_8}_{S_2} \mid \dots \mid \underbrace{s_{z-3} s_{z-2} s_{z-1} s_z}_{S_\beta}$ .  Write $S$ as a sequence $S'$ of matrices with $a_i \hat{=} V_j \iff j \equiv i \pmod{N}$ , it is $S' = \underbrace{V'_1 V'_2 V'_3 V'_4}_{S'_1} \mid \underbrace{V'_5 V'_6 V'_7 V'_8}_{S'_2} \mid \dots \mid \underbrace{V'_{z-3} V'_{z-2} V'_{z-1} V'_z}_{S'_\beta}$ , with $V'_i \in \{V_1, V_2, \dots, V_{2N}\}$ .  Calculate $\beta = \frac{z}{4}$ automorphisms $g_i \in \mathcal{G}_{Aut}$ . Compute $y_1 = \bar{\alpha}, y_2 = h(y_1), \dots, y_\beta = h(y_{\beta-1})$ and obtain $g_{y_1}, g_{y_2}, \dots, g_{y_\beta}$ .  For each part $S'_i$ , $1 \leq i \leq \beta$ , choose an additional matrix $P_i \in SL(2, \mathbb{Q})$ , with $P_i \notin F_{U'}$ , which is an ephemeral key.  Encryption: For $S'_i$ , $1 \leq i \leq \beta$ , choose ephemeral key $P_i$ and apply automorphism $g_{y_i}$ : $(V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i)$ $\downarrow g_{y_i}$ $(W_{5i-4}, W_{5i-3}, W_{5i-2}, W_{5i-1}, W_{5i})$ .  Generate ciphertext  $C = W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} \dots W_{z+\beta}$  and send it to Bob.	$C = W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} \dots W_{z+\beta}$ $\longrightarrow$

Table 1.14.: Summary of **Protocol 10**: Private key cryptosystem using automorphisms on plaintext sequences II

Alice	Bob
	Decryption
	<p>Cut <math>C</math> into parts of 5 matrices:  <math display="block">C = \underbrace{W_1W_2W_3W_4W_5}_{C'_1} \mid \underbrace{W_6W_7W_8W_9W_{10}}_{C'_2} \mid \cdots \mid \underbrace{W_{z+\beta-4} \cdots W_{z+\beta}}_{C'_\beta}.</math></p> <p>Compute <math>\beta</math> automorphisms:  <math>y_1 = \bar{\alpha}, y_2 = h(y_1), \dots, y_\beta = h(y_{\beta-1}),</math>  obtain <math>g_{y_1}, g_{y_2}, \dots, g_{y_\beta}.</math></p> <p>Compute for each automorphism <math>g_{y_i} \in \mathcal{G}_{Aut}, i = 1, 2, \dots, \beta,</math> the inverse automorphism <math>g_{y_i}^{-1}.</math></p> <p>Apply on each ciphertext part <math>C'_i</math> the corresponding automorphism <math>g_{y_i}^{-1}.</math> In general, for <math>C'_i,</math> it is:  <math display="block">(W_{5i-4}, W_{5i-3}, W_{5i-2}, W_{5i-1}, W_{5i})</math> <math display="block">\downarrow g_{y_i}^{-1}</math> <math display="block">(V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i)</math></p> <p>Decide which matrices in the reconstructed part belong to the set <math>U'</math> and which not.</p> <p>Therefore, get sequence of matrices</p> $S' = V'_1V'_2V'_3V'_4V'_5V'_6V'_7V'_8 \cdots V'_{z-3}V'_{z-2}V'_{z-1}V'_z,$ <p>with <math>V'_i \in \{V_1, V_2, \dots, V_{2N}\},</math> and with the knowledge</p> $a_i \hat{=} V_j \iff j \equiv i \pmod{N}$ <p>read the plaintext</p> $S = s_1s_2s_3s_4s_5s_6s_7s_8 \cdots s_{z-3}s_{z-2}s_{z-1}s_z$ <p>from Alice.</p>

We also look at the situations, when  $z \in \mathbb{N}$  is not necessarily  $z = 4 \cdot t, t \in \mathbb{N},$  and the letters in each part  $S_i$  are not necessarily pairwise different (Remark 9.0.2).

Alice has to calculate ephemeral matrices in  $SL(2, \mathbb{Q}),$  we also give proposals how Alice could generate these matrices, which are not matrices in  $F_{U'}$  (Remark 9.0.3).

**Security:**

**Protocol 10** is a polyalphabetic system, that is, a matrix  $V_i \in U',$  and hence a letter  $a_i \in A,$  is encrypted differently at different positions in the plaintext. Therefore, a statistical frequency attack (see for instance [BFKR15]), for the ciphertext, over the frequency of matrices, which corresponds to letters in the plaintext alphabet, or groups of words, is useless. If Alice takes

care, that each plaintext sequence  $C_i$  has as one product at least one ephemeral matrix  $P_j$  (that is, it is not only written as a word in  $U'$ ), then  $C_i \notin F_{U'}$  and each plaintext sequence is encrypted differently, even if the same text with the same automorphisms is used and only the ephemeral matrices from Alice are changed, she generates a totally different ciphertext for the same plaintext. The security depends on the ephemeral keys (matrices), which are privately chosen by the encrypter and the unknown solution of the (constructive) membership problem for the matrices in  $U' \subset \text{SL}(2, \mathbb{Q})$  and the ephemeral keys.

**Variations:**

We explained **Protocol 10** by fixing the rank of the abstract group  $G$ ,  $\text{rank}(G) = 5$ , and the cardinality of  $U$ ,  $|U| = 2N$ . This helped us to explain this private key cryptosystem, but it is not mandatory. We can variate these values.

1. We can choose an abstract free group  $G$  with finite  $\text{rank}(G) \geq 2$ ; another option is to choose different free groups  $G_i$  with pairwise different finite ranks  $\text{rank}(G_i) \geq 2$ . The set  $\mathcal{G}_{Aut}$  from which Alice and Bob get the automorphisms for encryption and decryption, respectively, is then a subset of  $\bigcup_i \text{Aut}(G_i)$ .
2. The set  $U$  can be chosen with cardinality  $|U| = k \cdot N$ ,  $k \geq 2$ . Each sequence  $S_i$ , which we get by cutting the plaintext into pieces, must, in general, have a length between 1 and  $\text{rank}(G_j) - 1$ , depending on the group  $G_j$  on which the automorphism acts for the sequence  $S_i$ , because we now add to each sequence at least one ephemeral matrix. We have to take care that the elements in each sequence  $S_i$  with the additional ephemeral keys form a basis for a free group of  $\text{rank}(G_j)$ . Therefore, it is possible to align the set  $U$ , that is, choose  $k$  with  $k \geq \max\{\text{rank}(G_i)\} - 1$  for  $|U| = k \cdot N$  and act like in Remark 9.0.2 case 2.1. Another option is to split the sequence with more than  $k$  identical letters in a similar way as explained in Remark 9.0.2 case 2.2. The ephemeral keys in each sequence must be pairwise different and after construction, see Remark 9.0.3, they are elements of a basis.

We present also a variation, in which the used automorphisms are private.

**Conclusion concerning chosen plaintext attacks and chosen ciphertext attacks:**

This cryptosystem is secure against chosen plaintext and chosen ciphertext attacks, because of the ephemeral keys by Alice and the unknown solution of the (constructive) membership problem for the matrices in  $U' \subset \text{SL}(2, \mathbb{Q})$  and the ephemeral keys.

**Conclusion:**

**Protocol 10** applies automorphisms on sequences of plaintext units, which are written as matrices in  $\text{SL}(2, \mathbb{Q})$  with the help of a faithful representation from a finitely generated free group  $F$  into  $\text{SL}(2, \mathbb{Q})$ . With additional random matrices for the encryption procedure and the unknown solvability of the (constructive) membership problem in the used matrix group it is a secure private key cryptosystem. Thus, we get another interesting new private key cryptosystem, which uses non-commutative groups and is based on combinatorial group theory and not on number theory.

**Chapter 10**

Chapter 10 introduces **Protocol 11**, an ElGamal like public key cryptosystem, and **Protocol 12**, a challenge and response system. **Protocol 11** is published in [MR15] and [MR16]. Both systems are based on combinatorial group theory and use the ideas behind the private key cryptosystems in the previous sections. Thus, they also need a finitely generated free group  $F$ ,

automorphisms on  $F$  and a faithful representation from  $F$  into  $SL(2, \mathbb{Q})$  (see Chapter 4).

**Inner outline of Chapter 10:**

Firstly, we describe **Protocol 11** and give proposals for variations and an example. Secondly, **Protocol 12** is introduced and an example can be found in the appendix.

- **Protocol 11:** ElGamal like public key cryptosystem using automorphisms on a finitely generated free group  $F$

**Protocol 11**, an ElGamal like cryptosystem, is summarized in Table 1.15 (page 52), which is exactly Table 10.1 (page 236) in Chapter 10.

Table 1.15.: Summary of **Protocol 11:** ElGamal like public key cryptosystem using automorphisms on a finitely generated free group  $F$

<b>Public Parameters</b>	
Free group $F = \langle X \mid \ \rangle$ , a freely reduced word $a \neq 1$ in $F$ and an automorphism $f : F \rightarrow F$ of infinite order.	
Alice	Bob
Key Creation	
Choose private key $n \in \mathbb{N}$ . Compute $f^n(a) =: c \in S^*.$ ( $S^*$ denotes the set of all freely reduced words with letters in $X^{\pm 1}$ .) Publish $c$ .	
Encryption	
	Choose plaintext $m \in S^*$ . Choose random ephemeral key $t \in \mathbb{N}$ . Compute $m \cdot f^t(c) =: c_1 \in S^* \quad \text{and} \quad f^t(a) =: c_2 \in S^*.$ Send ciphertext $(c_1, c_2) \in S^* \times S^*$ to Alice. $\xleftarrow{(c_1, c_2)}$
Decryption	
Compute $\begin{aligned} c_1 \cdot (f^n(c_2))^{-1} &= m \cdot f^t(c) \cdot (f^n(c_2))^{-1} \\ &= m \cdot f^t(f^n(a)) \cdot (f^n(f^t(a)))^{-1} \\ &= m \cdot f^{t+n}(a) \cdot (f^{n+t}(a))^{-1} \\ &= m, \end{aligned}$ which is the message from Bob.	

**Security:**

The security is based on the Diffie-Hellman problem and discrete logarithm problem in cyclic subgroups of automorphisms on finitely generated free groups.

**Variations:**

1. The element  $a \in S^*$  could be taken as a common private secret between Alice and Bob. They could use for example the Anshel-Anshel-Goldfeld key exchange protocol (see [MSU08]) to agree on the element  $a$ .
2. Alice and Bob agree on a faithful representation from  $F$  into the special linear group of all  $2 \times 2$  matrices with entries in  $\mathbb{Q}$ , that is,  $g : F \rightarrow \text{SL}(2, \mathbb{Q})$ . Now,  $m \in S^*$  and Bob sends  $g(m) \cdot g(f^t(c)) =: c_1 \in \text{SL}(2, \mathbb{Q})$  instead of  $m \cdot f^t(c) =: c_1 \in S^*$ ;  $c$  and  $c_2$  remain the same. Therefore, Alice calculates  $c_1 \cdot (g(f^n(c_2)))^{-1} = g(m)$  and hence the message  $m = g^{-1}(g(m)) \in S^*$ . This variation in addition extends the security certification to the constructive membership problem in the matrix group  $\text{SL}(2, \mathbb{Q})$  (see [EKL14]).

**Conclusion:**

**Protocol 11** is similar to the ElGamal cryptosystem (see Section 1.2.2), whereby the ElGamal cryptosystem is easier to handle. The ElGamal cryptosystem is based on the Diffie-Hellman problem and discrete logarithm problem, respectively, over a finite field. If these problems should eventually be solved we introduced an alternative system, which is not based on number theory.

- **Protocol 12:** Challenge and response protocol using automorphisms on finitely generated free groups

We use the idea behind the public key cryptosystem (**Protocol 11**) based on Nielsen transformations to develop a challenge and response protocol. More precisely **Protocol 12** is a symmetric key authentication protocol.

The verifier and the prover agree on a common secret  $P$  and a corresponding challenge automorphism of a finitely generated free group  $F$  with rank  $N \geq 3$ . After presenting the password  $P$  to the verifier, he gives challenges to the prover, which are correctly solvable with the challenge automorphism.

**Possible Challenges:**

We propose four types of questions for the challenges.

1. What is the matrix  $M = \varphi(f^n(w))$ , given  $w \in F$  (a freely reduced word in  $F$ ),  $n \in \mathbb{N}$  and a faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ ? The verifier takes care that each matrix  $\varphi(x_i)$ ,  $1 \leq i \leq N$ , has at least one entry in  $\mathbb{Q} \setminus \mathbb{Z}$ .
2. What is the trace of the matrix  $M = \varphi(f^n(w))$ , given  $w \in F$  (a freely reduced word in  $F$ ),  $n \in \mathbb{N}$  and a faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ ? The verifier takes care that each matrix  $\varphi(x_i)$ ,  $1 \leq i \leq N$ , has at least one entry in  $\mathbb{Q} \setminus \mathbb{Z}$ .
3. What is the entry  $M_{x,y}$  of the matrix  $M = \varphi(f^n(w))$ , given  $w \in F$  (a freely reduced word in  $F$ ),  $n \in \mathbb{N}$  and a faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ , with  $x, y \in \{1, 2\}$  and  $x$  gives the row and  $y$  the column in the matrix  $M$ ? A variation could be given if the entry  $M_{x,y}$  is an integer, then it could be ask for certain digits of an entry  $M_{x,y}$ , for example for the last 7 digits.

4. Questions as in 1. and 2. but the faithful representation  $\varphi$  could be public or also a part of the common shared secret between the verifier and the prover.

**Protocol 12** is summarized in Table 1.16 (page 54), which is exactly Table 10.2 (page 241) in Chapter 10.

Table 1.16.: Summary of **Protocol 12**: Challenge and response protocol using automorphisms on finitely generated free groups

<b>Private Parameters</b>	
Free group $F$ with free generating set $X = \{x_1, x_2, \dots, x_N\}$ , $N \geq 3$ ; an automorphism $f \in \text{Aut}(F)$ of infinite order and a common password $P$ . The shared secret is the tuple $(P, f)$ .	
<b>Verifier</b>	<b>Prover</b>
	Present the password $P$ to the verifier $P$
←	
Take challenge automorphism $f$ corresponding to password $P$ . Choose <ul style="list-style-type: none"> <li>• a faithful representation <math>\varphi : F \rightarrow SL(2, \mathbb{Q})</math>;</li> <li>take care that each matrix <math>\varphi(x_i)</math>, <math>1 \leq i \leq N</math>, has at least one entry in <math>\mathbb{Q} \setminus \mathbb{Z}</math>;</li> <li>• a freely reduced word <math>w \in F</math>;</li> <li>• <math>n \in \mathbb{N}</math>.</li> </ul>	
Challenge: $(\varphi, w, n)$ →	
Compute $M' = \varphi(f^n(w))$ .	Compute the response $M$ and send it to the verifier  $M = \varphi(f^n(w))$ .
← Response: $M$	
Proof if $M' = M$ .	

**Security:**

The Security is based on the unknown solution of the constructive membership problem of (discrete) free subgroups in  $SL(2, \mathbb{Q})$  of rank greater than two.

**Conclusion:**

To develop a challenge and response system using automorphisms on finitely generated free groups is another cryptologic application of this mathematical theory. It is good to have different challenge “spaces” for challenge and response systems, in particular if these “spaces” generate an infinite amount of challenges as it is the case in **Protocol 12**.

**1.3.3. Assessment of the results**

The first private key cryptosystem (**Protocol 1**) is an extension of a  $(n, t)$ -secret sharing scheme by C. S. Chum, B. Fine, G. Rosenberger and X. Zhang (CFRZ-scheme) and is therefore based on the Closest Vector Theorem. This extension is of mathematical interest. However, for applications it has the disadvantage that Alice and Bob have to exchange the subspace  $V$  not later

than after  $t$  messages, with  $\dim(V) = t$ . This private key cryptosystem, along with other cryptographic theory, was published in [FMR13] and presented in the “Algebra and Cryptography” seminar at CUNY (City University of New York) Graduate Center in New York in 2013 as well as in the “Spring Easter Section Meeting” of the AMS (American Mathematical Society) more precisely the “Special session on ‘Algorithmic problems of group theory and applications to information security’ at Boston College” in 2013 by an invitation from Prof. Dr. V. Shpilrain and Prof. Dr. D. Kahrobaei. The positive feedback gave motivation to find another cryptographic protocol based on the Closest Vector Theorem. This is the CV-challenge and response system (**Protocol 2**).

In this thesis, we also introduce two extensions of a key exchange protocol by M. Habbeb, D. Kahrobaei, C. Koupparis and V. Shpilrain, which is based on a semidirect product of (semi)groups. One extension is a signature protocol (**Protocol 3**) and the other is an ElGamal like public key cryptosystem (**Protocol 4**). Both are of mathematical interest and published in [Mol15]. For applications of the signature protocol it has the disadvantage that Alice, with her private key  $n$ , can just perform a finite number of signatures, which depend on her first ephemeral key  $k_1$ . The theory of the ElGamal like public key cryptosystem is interesting, but for applications a platform group has to be found, which is optimal in terms of security and efficiency. D. Kahrobaei and V. Shpilrain are working on this problem, see [KS16].

The research about these cryptographic protocols leads to groups and especially to non-commutative groups, which gives input for the later newly developed cryptographic protocols that are based on combinatorial group theory.

Two new challenge and response systems are explained. **Protocol 2** is an extension of the CFRZ-scheme and a variation is given, such that it is a two-way authentication. **Protocol 12** is one of the newly developed cryptographic protocols which are based on combinatorial group theory. It is good to get different challenge “spaces” for challenge and response systems, in particular, if these “spaces” generate an infinite amount of challenges as it is the case for these two cryptographic protocols. **Protocol 2** was, among other protocols, presented at the workshop “New directions in cryptography” related to the “Computer Science in Russia” symposium in Moscow (2014).

**Protocol 5** is a secret sharing protocol, which is highly interesting if the participants need a very simple way to reconstruct the secret and the dealer has enough time to generate and to distribute the shares for the participants.

**Protocol 6** and **Protocol 7** are the first newly developed cryptographic protocols in this thesis, which use combinatorial group theory especially Nielsen transformations and finitely generated free groups. **Protocol 6** is mathematically a highly interesting cryptographic protocol. Indeed it serves especially well as a basis to develop other cryptographic protocols. In this thesis it is the basis for **Protocol 7-12**, which are also based on combinatorial group theory.

**Protocol 7** is, like **Protocol 6**, mathematically a very interesting cryptographic protocol which in addition uses a Nielsen reduced subset  $U \neq X$  of a finitely generated free group  $F = \langle X \mid \ \ \rangle$  and gives therefore the final input for the newly developed cryptographic **Protocol 8-12**.

**Protocol 5** and **Protocol 6**, as **Protocol 2**, were presented at the workshop “New directions in cryptography” related to the “Computer Science in Russia” symposium in Moscow (2014). Enhancing the combinatorial secret sharing scheme with the idea of using automorphisms on finitely generated free groups became the basis for the other newly developed cryptographic protocols (**Protocol 8-12**). Moreover, **Protocol 5**, **Protocol 6** and **Protocol 7** were also presented at the Fairfield University (2015) by an invitation from Prof. Dr. B. Fine and they

are going to be published in the survey article [CFMRZ16] as research in the area of secret sharing schemes. They are also described in [MR15].

In addition **Protocol 11** and a previous version of **Protocol 9** are presented in [MR15].

The main results in this thesis are, in addition to the challenge and response system (**Protocol 12**), the private key cryptosystems **Protocol 8**, **Protocol 9**, **Protocol 10** and the ElGamal like public key cryptosystem **Protocol 11**, which are all based on combinatorial group theory and automorphisms on finitely generated free groups. These automorphisms can be generated by Nielsen transformations or Whitehead-Automorphisms.

Especially the modification of **Protocol 8** and **Protocol 9**, respectively, with matrices are of interest for group based cryptography. If **Protocol 8** or **Protocol 9**, respectively, is used together with a modification, which uses a faithful representation from a finitely generated free group into  $SL(2, \mathbb{Q})$ , then the system is secure and the security depends on the unknown solution of the (constructive) membership problem in the used matrix group. If **Protocol 8** or **Protocol 9**, respectively, is used together with the modification, which uses matrices in  $GL(2, R)$ ,  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ ,  $n \geq 2$ , then the system is secure and the security depends in addition on the negative solution of Hilbert's Tenth Problem. Moreover, we get also randomness to each ciphertext by the ephemeral matrices which were used for encryption. To generate these ephemeral matrices only the common secret point  $D \in \mathbb{Z}^n$  is needed, this improves also the security. In fact we get two interesting new private key cryptosystems, which use non-commutative groups and are based on combinatorial group theory and not only on number theory.

**Protocol 10** applies automorphisms on sequences of plaintext units, which are written as matrices in  $SL(2, \mathbb{Q})$ . Together with additional random matrices for the encryption procedure and the unknown solvability of the (constructive) membership problem in the used matrix group it is a secure private key cryptosystem. Thus, we get another interesting new private key cryptosystem, which uses non-commutative groups and is based on combinatorial group theory and not on number theory.

**Protocol 11** is similar to the standard ElGamal cryptosystem; however the standard ElGamal cryptosystem is easier to handle. The ElGamal cryptosystem is based on the Diffie-Hellman problem and discrete logarithm problem, respectively, over a finite field. If these problems should eventually be solved we introduce here an alternative system, which is not based on number theory.

A previous version of **Protocol 8** together with **Protocol 11** was presented at the annual meeting of the DMV (Deutsche Mathematiker-Vereinigung) at the minisymposium "Algebraic Aspects of Cryptology" in 2015. Following an invitation from Prof. Dr. V. Shpilrain a talk was held about the research in the "Algebra and Cryptography" seminar at CUNY Graduate Center in New York. **Protocol 8** together with the modification, which uses a faithful representation of a finitely generated free group into  $SL(2, \mathbb{Q})$ , as well as **Protocol 10**, were presented in this seminar at CUNY Graduate Center in New York (2015). In addition **Protocol 8** together with the modification using a faithful representation of a free group into  $SL(2, \mathbb{Q})$  and the idea to use Hilbert's Tenth Problem are published in [MR16] together with **Protocol 11**.

## 1.4. Suggestions for other platform groups instead of finitely generated free groups

In this section we give other platform groups instead of finitely generated free groups for the cryptographic protocols **Protocol 6** to **Protocol 12**.

We suggest the use of surface groups. A surface group is the fundamental group of a compact



orientable or non-orientable surface. It is defined, that the genus of a surface group is  $g$  if the corresponding surface has genus  $g$  (for surface groups in combinatorial group theory see for example [AFR05] and [FR99]). It is known (see for instance [BS65]), that an orientable surface group  $\Phi_g$  of genus  $g \geq 2$  has an one-relator presentation of the form

$$\Phi_g = \langle \alpha_1, \beta_1, \dots, \alpha_g, \beta_g \mid \prod_{j=1}^g [\alpha_j, \beta_j] = 1 \rangle, \quad (1.1)$$

in which  $[\alpha_j, \beta_j]$  denotes the commutator of  $\alpha_j$  and  $\beta_j$  and is defined as  $[\alpha_j, \beta_j] := \alpha_j \beta_j \alpha_j^{-1} \beta_j^{-1}$ . Depending on the new introduced cryptographic protocols **Protocol 6** to **Protocol 12** Alice and Bob need subgroups of the used group, a basis of this subgroup, a faithful representation into  $\mathrm{SL}(2, \mathbb{Q})$  or  $\mathrm{PSL}(2, \mathbb{Q})$ , respectively, and automorphisms (or Nielsen transformations, which in the free group case describe the automorphisms of the free group) of the group or subgroup, respectively.

Magnus shows in [Mag73] that  $\Phi_g$ , for  $g \geq 2$ , has infinitely many faithful representations as a discontinuous subgroup of  $\mathrm{PSL}(2, \mathbb{Q})$ . He obtains this result by finding 2 by 2 matrices  $\alpha$  and  $\beta$  whose elements are rational functions of two parameters  $r$  and  $t$ , such that  $\alpha$  and  $\beta$  generate a faithful representation of the group  $G$ , defined by

$$G = \langle \alpha, \beta \mid [\alpha, \beta]^2 = 1 \rangle, \quad (1.2)$$

for all  $r > 1$ ,  $t > 0$ , and every Fuchsian group isomorphic with  $G$  conjugates in  $\mathrm{PSL}(2, \mathbb{R})$  with one of these representations.

He proves the Theorem 1.4.1, which is needed to get a representation of  $\Phi_2$  as discontinuous group of  $\mathrm{PSL}(2, \mathbb{Q})$ .

**Theorem 1.4.1.** [Mag73, Theorem 1]

Let  $r, t$  be real parameters and let  $r^2 > 1$ ,  $t \neq 0$ . Then the matrices

$$\alpha = \begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix} \quad \text{and} \quad \beta = \begin{pmatrix} \frac{1}{t(r^2-1)} & tr^2 \\ \frac{2}{t(r^2-1)^2} & \frac{t(1+r^4)}{r^2-1} \end{pmatrix}$$

generate a discontinuous subgroup of  $\mathrm{PSL}(2, \mathbb{R})$  isomorphic with  $G$  as defined by (1.2). Every discontinuous subgroup of  $\mathrm{PSL}(2, \mathbb{R})$  isomorphic with  $G$  is conjugated in  $\mathrm{PSL}(2, \mathbb{C})$  with one of the groups generated by  $\alpha, \beta$  for suitable values of  $r, t$ .

With the following lemma, he gives a construction to generate  $\Phi_2$  with the help of  $G$  defined by (1.2) and  $\Phi_2$  contains  $\Phi_g$  for  $g \geq 2$  as a subgroup of index  $g - 1$ .

**Lemma 1.4.2.** [Mag73, Lemma 2]

Given  $\alpha$  and  $\beta$  as in Theorem 1.4.1. Define the elements  $\alpha_1, \beta_1, \alpha_2, \beta_2, \gamma$  of  $G$  by

$$\alpha_1 := \alpha^2, \quad \beta_1 := \beta, \quad \alpha_2 := \gamma^{-1} \alpha^2 \gamma, \quad \beta_2 := \gamma^{-1} \beta \gamma, \quad \gamma := \alpha \beta \alpha^{-1} \beta^{-1}.$$

Then  $\alpha_1, \beta_1, \gamma$  generate a subgroup  $G_2$  of index 2 in  $G$  with defining relations

$$(\alpha_1 \beta_1 \alpha_1^{-1} \beta_1^{-1} \gamma^{-1})^2 = 1 \quad \text{and} \quad \gamma^2 = 1,$$

and  $\alpha_1, \beta_1, \alpha_2, \beta_2$  generate a subgroup of index 4 in  $G$  which is isomorphic with  $\Phi_2$  (defined by (1.1) for  $g = 2$ ). The group  $\Phi_2$  contains  $\Phi_g$  for  $g \geq 2$  as a subgroup of index  $g - 1$ .

The proof of this lemma consists of an application of the Reidemeister-Schreier-Method (see [CgRR08] or [MKS66] or [LS77]). The group  $G_2$  has coset representatives  $1, \alpha$  in  $G$  and  $\Phi_2$

has coset representatives  $1, \gamma$  in  $G_2$ . For  $\Phi_g$  we may choose the normal closure of the subgroup generated by  $\alpha_1^{g-1}$  and  $\beta_1, \alpha_2, \beta_2$ .

**Corollary 1.4.3.** [Mag73, Corollary 1.2.]

*Whenever  $r^2 > 1$  and  $t \neq 1$  are rational numbers, the representation of  $G$  in Theorem 1.4.1 produces a representation of  $\Phi_2$  as discontinuous subgroup of  $\text{PSL}(2, \mathbb{Q})$ .*

If Alice and Bob choose the rational parameters  $r$  and  $t$ , with  $r^2 > 1$  and  $t \neq 1$  (see Corollary 1.4.3 and Lemma 1.4.2), they get a faithful representation for  $\Phi_2$  in  $\text{PSL}(2, \mathbb{Q})$  and if they change the parameters they change the faithful representation for  $\Phi_2$  in  $\text{PSL}(2, \mathbb{Q})$ .

Alice and Bob would use the orientable surface group  $\Phi_2$  instead of the finitely generated free group  $F = \langle x_1, x_2, \dots, x_g \mid \ \rangle$  and with Lemma 1.4.2 they are able to generate each subgroup  $\Phi_g$ , for  $g \geq 2$ , in  $\Phi_2$ , which they use instead of  $F_U$ . The  $2g$  generators of this subgroup are the basis for the subgroup, because fewer than  $2g - 1$  elements in  $\Phi_g$  generate a free group (see [FR99, p. 54]), hence the set of  $2g$  generators of  $\Phi_g$  is a minimal generating set for  $\Phi_g$ , thus a basis.

It is possible to use Nielsen transformations for the cryptosystems because orientable surface groups have only one Nielsen class of minimal generating systems:

Let  $H = \langle a_1, a_2, \dots, a_n \mid r = 1 \rangle$  be a one-relator group with  $m \geq 2$  and  $r$  a relation which is a cyclically reduced word and involves all elements  $a_1, a_2, \dots, a_n$ . Let  $H$  not be a free group, that is,  $r \neq 1$  and  $r$  is not a primitive element in  $F = \langle a_1, a_2, \dots, a_n \mid \ \rangle$ . We say  $H$  has exactly **one Nielsen class**, if each generating system  $\{x_1, x_2, \dots, x_n\}$  of  $H$  is Nielsen equivalent to  $\{a_1, a_2, \dots, a_n\}$  (see [CgRR08]).

Alice and Bob are able to execute the cryptographic protocols above with orientable surface groups  $\Phi_g$  and Nielsen transformations, because orientable surface groups  $\Phi_g$ ,  $g \geq 2$ , have exactly one Nielsen class (see [CgRR08]) and a faithful representation into  $\text{PSL}(2, \mathbb{Q})$ .

We now give two remarks about two further platform groups, which could be of interest for the cryptographic protocols.

**Remark 1.4.4.** A presentation of a non-orientable surface group  $N_g$  of genus  $g \geq 2$  can be given by the following one-relator presentation

$$N_g = \langle \alpha_1, \alpha_2, \dots, \alpha_g \mid \alpha_1^2 \alpha_2^2 \cdots \alpha_g^2 = 1 \rangle,$$

see [FR99] and the references there. For the non-orientable surface groups with genus  $g \geq 4$  it is known that these have exactly one Nielsen class of minimal generating systems (see [CgRR08]), and that there is a faithful representation as discrete subgroup of  $\text{PSL}(2, \mathbb{R})$  (see [FKR14]). Thus, we could use these groups for the cryptosystems above but we are then in the case in which we store and calculate real numbers. In cryptography it is more required to use elements not in  $\mathbb{R} \setminus \mathbb{Q}$  thus we would like to get a faithful representation into the group  $\text{PSL}(2, \mathbb{Q})$  but it is unknown if such an embedding exists.

**Remark 1.4.5.** For other platform groups we suggest finitely generated elementary free groups. Nonabelian groups that have exactly the same first order theory (see for example Appendix A.2 or [FGMRS14, Chapter 4]) as the class of nonabelian free groups are called elementary free groups (see for instance [FGMRS14, Chapter 10]). The primary non-free examples of such groups are orientable surface groups  $\Phi_g$  of genus  $g \geq 2$  and non-orientable surface groups  $N_g$  of genus  $g \geq 4$ .

The finitely generated elementary free groups are hyperbolic, see [FGMRS14, Theorem 10.4.1]. It is known, that any finitely generated elementary free group has a faithful representation

into  $\mathrm{PSL}(2, \mathbb{C})$  ([FGMRS14, Theorem 10.4.11]). It is also known, that any limit group can be embedded in  $\mathrm{PSL}(2, \mathbb{R})$  ([FGMRS14, Theorem 7.3.5]) and elementary free groups are limit groups ([FGMRS14, Section 10.4.4]), thus we get also a faithful representation into  $\mathrm{PSL}(2, \mathbb{R})$ . There is a conjecture that finitely generated hyperbolic elementary free groups have only one Nielsen class of minimal generating systems. As above, Remark 1.4.4, we could use a representation into  $\mathrm{PSL}(2, \mathbb{R})$  for the cryptosystem but for cryptography it is more required to use a faithful representation into  $\mathrm{PSL}(2, \mathbb{Q})$  but it is unknown if such an embedding exists.

## 1.5. Open questions and further research for cryptographic protocols based on combinatorial group theory

We give some ideas for further research questions.

- Try to find other cryptographic protocols, which can be based on Nielsen transformations, for example a public key cryptosystem which is not ElGamal like.
- Is there a faithful representation into the group  $\mathrm{PSL}(2, \mathbb{Q})$  for the non-orientable surface groups with genus  $g \geq 4$ ?
- For finitely generated elementary free group is there a faithful representation into  $\mathrm{PSL}(2, \mathbb{Q})$ ?
- We developed symmetric key cryptosystems which use abstract presentations of free groups and improve them by using the special linear group  $\mathrm{SL}(2, \mathbb{Q})$  for the presentation of the used group. The additional security certification is now, that there is no algorithm known to solve the membership problem for (discrete) free subgroups of  $\mathrm{SL}(2, \mathbb{Q})$ . B. Eick, M. Kirschmer and C. Leedham-Green presented in the paper [EKL14] a practical algorithm to solve the constructive membership problem for (discrete) free subgroups of rank 2 of  $\mathrm{SL}(2, \mathbb{R})$ . For example, the subgroup  $\mathrm{SL}(2, \mathbb{Z})$  of  $\mathrm{SL}(2, \mathbb{R})$  is discrete. They also mention, that it is an open problem to solve the membership problem for arbitrary subgroups of  $\mathrm{SL}(2, \mathbb{R})$  with rank  $m \geq 2$ . The developed cryptosystems work with exactly such subgroups of rank greater than or equal to 2. For further research one could work on the solvability of the membership problem for arbitrary subgroups of  $\mathrm{SL}(2, \mathbb{R})$  with rank  $m \geq 2$ . B. Eick, M. Kirschmer and C. Leedham-Green presented a practical algorithm to solve the constructive membership problem for discrete free subgroups of rank 2 of  $\mathrm{SL}(2, \mathbb{R})$ , hence it is convenient to start with the analysis of discrete groups of rank greater than 2 and the constructive membership problem, therefore it is useful to study the algebraic generalizations of discrete groups (see for example [FR99]).
- A lot of open questions exist, especially in the non-commutative group based cryptography, which are interesting for research, for example (see [FHKR11]):
  1. What is the most appropriate platform group for non-commutative cryptography?
  2. Should the group be finite or infinite?
  3. How can we show a group is provably secure for the new non-commutative schemes such as public key exchanges, signatures, authentication protocols et cetera?
  4. Can we design more public keys based on other search and decision problems in combinatorial group theory? Can we analyze the security of this cryptographic protocols?
  5. What should be the measure of the security? (Practicality, complexity, average case complexity, generic complexity?)

6. What is the complexity of the Reidemeister-Schreier rewriting algorithm for free groups?

In [FHKR11] B. Fine, M. Habeeb, D. Kahrobaei and G. Rosenberger mention also that another problem to think in this direction is quantum computational approaches to cryptosystems. Quantum algorithms for finite solvable groups (which are polycyclic) have been studied, particularly by J. Watrous [Wat00]. He found a quantum algorithm to compute the order of a finite solvable group in polynomial time. The algorithm works in the setting of black-box groups none of them having polynomial-time classical algorithms. Is it possible to design quantum algorithms for solving other decision problems in polycyclic groups (both for finite and infinite ones); especially the ones which are used in cryptography? The following questions appear:

1. Is there any quantum algorithm for solving the search conjugacy problem for polycyclic groups that reduces the complexity of the algorithm?
2. Are there other quantum algorithms for problems in combinatorial group theory?

## Acknowledgements

I acknowledge my debt of thanks to my advisor Gerhard Rosenberger, who greatly improved my knowledge of mathematics and for his outstanding support and encouragement during the last years.

I am very grateful to my co-advisor Ulf Kühn for his excellent support during my studies and my Ph.D. project.

My special thank is dedicated to Benjamin Fine, with whom I enjoyed to collaborate. Especially, his invitation to Fairfield with a lot of fruitful discussions enriches my research.

I would like to thank Delaram Kahrobaei and Vladimir Shpilrain for giving me the opportunity to present my research during their conferences and seminars wherein I had the chance to meet some outstanding mathematicians in the field of my research.

# Chapter 2

## Inner product spaces and cryptography

In this chapter we introduce **Protocol 1** and **Protocol 2**, which extend the CFRZ-secret sharing scheme to a private key cryptosystem as well as to a challenge and response system. The CFRZ-scheme is a  $(n, t)$ -secret sharing protocol, which is based on the Closest Vector Theorem in a real inner product space (see Theorem 2.0.1). The idea behind the CFRZ-scheme was first published by C. S. Chum, B. Fine, G. Rosenberger and X. Zhang in [CFRZ12]. It was worked out and analyzed in detail in [Mol12] whereby parts of these results were published in [FMR13] and the overview article [CFMRZ16]. We require knowledge of linear algebra and analytic geometry, as it is presented for example in the books [Bos08] or [Fis10].

**Protocol 1** is a private key cryptosystem, which is published in [FMR13]. We call it also CV-private key cryptosystem, because it uses the **Closest Vector Theorem**.

**Protocol 2** is a challenge and response system, which is also named CV-challenge and response protocol, because it also makes use of the **Closest Vector Theorem**.

First of all we recall the Closest Vector Theorem and the CFRZ-scheme. Afterwards we introduce **Protocol 1** and **Protocol 2** in detail.

All cryptographic protocols in this chapter require the following theorem.

**Theorem 2.0.1.** [Atk89] Closest Vector Theorem

*Let  $W$  be a real inner product space and let  $V$  be a subspace of finite dimension  $t$ ,  $t \in \mathbb{N}$ . Suppose that  $w^* \in W$ , with  $w^* \notin V$ , and  $e_1, e_2, \dots, e_t$  is an orthonormal basis of  $V$ . Then the unique vector  $w \in V$  closest to  $w^*$  is given by*

$$w = \langle w^*, e_1 \rangle e_1 + \langle w^*, e_2 \rangle e_2 + \dots + \langle w^*, e_t \rangle e_t,$$

where  $\langle \cdot, \cdot \rangle$  is the inner product on  $W$ .

Let  $W$  be a real inner product space and  $V \subset W$  with  $\dim(W) = m$  and  $\dim(V) = t$ ,  $m > t$ . We denote the element in  $V$  which is the closest element to  $w^* \in W \setminus V$  with  $w$ .

A situation where  $W = \mathbb{R}^3$  and  $V$  is a two dimensional subspace, which is indicated by a yellow area, is visualized in Figure 2.1.

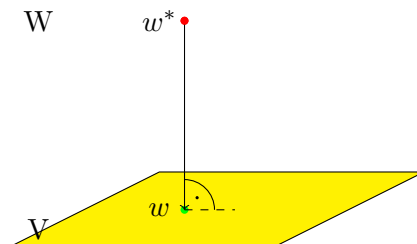


Figure 2.1.: Visualization of a situation in a CFRZ-scheme

Now, we give the idea of the CFRZ-secret sharing protocol. Afterwards we explain it in more details. The dealer determines the numbers  $n, t \in \mathbb{N}$ , with  $n \geq t$ . He chooses a real inner product space  $W$  of dimension  $m$  and a subspace  $V \subset W$  of dimension  $t$ , with

$$t = \dim(V) \quad \text{and} \quad m = \dim(W) > t.$$

The secret is an element  $w \in V$ . The dealer calculates an element  $w^* \in W \setminus V$ , such that  $w$  is the closest element in  $V$  to  $w^*$ , that is,  $\|w^* - w\| \leq \|w^* - v\|$  for all  $v \in V$ , here  $\|\cdot\|$  denotes the euclidean norm in  $W$ . The dealer constructs a set  $M = \{v_1, v_2, \dots, v_n\}$  with the property, that arbitrary  $t$  elements of  $M$  form a basis for the subspace  $V$ . The participant  $p_i$  receives the share  $v_i$ , for  $1 \leq i \leq n$ . The element  $w^*$  is sent to each participant or is published. If  $t$  or more participant join their shares, they are able to reconstruct the secret  $w \in V$  with the help of the public element  $w^*$ .

We explain in more details the steps for a  $(n, t)$ -secret sharing scheme, which are done by the dealer and the participants, following the paper [FMR13].

Steps for the dealer:

The integers  $n$  and  $t$  are given,  $n$  is the number of participants and  $t$  is the threshold and the dimension of the subspace  $V \subset W$ , respectively.

1. The dealer defines the dimension  $m \in \mathbb{N}$  of the real inner product space  $W$  with the property  $m > t$ .
2. He picks a secret  $w \in W$ .
3. Now, he chooses a subspace  $V \subset W$  with dimension  $t$ , such that  $w \in V$ .
4. He determines a set of vectors in  $V$  as  $M = \{v_1, v_2, \dots, v_n\}$  with the property that any subset of size  $t$  is independent. Hence, any such subset defines a basis for the subspace  $V$ .
5. The dealer calculates the closest vector  $w^* \in W \setminus V$ , such that  $w \in V$  is the closest vector in  $V$  to  $w^*$ , as follows:
  - a) He chooses a basis  $\{b_1, b_2, \dots, b_t\}$  of the subspace  $V$  and computes the orthogonal complement  $V^\perp = \{u \in W | u \perp v \text{ for all } v \in V\}$  to  $V$ .
  - b) Let  $B^\perp = \{b_1^\perp, b_2^\perp, \dots, b_{m-t}^\perp\}$  be a basis of the orthogonal complement  $V^\perp$ . Hence,  $w^*$  can be calculated as follows

$$w^* = \underbrace{w}_{\in V} + \underbrace{(\alpha_1 b_1^\perp + \alpha_2 b_2^\perp + \dots + \alpha_{m-t} b_{m-t}^\perp)}_{=: w^\perp \in V^\perp} \in W \setminus V,$$

with  $\alpha_i \in \mathbb{R}$  and at least one  $\alpha_i \neq 0$ , with  $1 \leq i \leq m - t$ .

6. The dealer distributes the vector  $v_i$  to the participant  $p_i$ , for all  $1 \leq i \leq n$ . The vector  $w^*$  is either sent to each participant or is published.

Steps for the participants:

If  $t$  out of  $n$  participants combine their parts, then the secret  $w$  can be recovered as follows:

1. The  $t$  vectors (shares from the participants) form a basis for the subspace  $V$  and hence using the Gram-Schmidt procedure together with a normalization (see for instance [Atk89]) they determine an orthonormal basis  $G = \{e_1, e_2, \dots, e_t\}$  of the subspace  $V$ .

2. Together with the vector  $w^*$  they are able to reconstruct the secret  $w$  with the help of Theorem 2.0.1, that is,

$$w = \langle w^*, e_1 \rangle e_1 + \langle w^*, e_2 \rangle e_2 + \cdots + \langle w^*, e_t \rangle e_t.$$

**Security 2.0.2.** If a subset of participants of size less than  $t$  is given, then these participants can only generate a subspace  $V'$  of dimension less than  $t$ . In the real inner product space  $W$  there are infinitely many extensions with dimension  $t$  to subspaces which contain the subspace  $V'$ . Hence, the probability to determine  $V$  from  $V'$  is negligible and it is very unlikely to reconstruct the secret  $w \in V$ . The probability to choose the correct subspace is negligible. Therefore, each possible secret is equally likely.

The public vector  $w^*$  tells an eavesdropper only the dimension of the real inner product space  $W$ . The eavesdropper does not know on which subspace he has to project the vector  $w^*$  to get the secret  $w$ , since infinitely many subspaces have to be considered for the projection.

**Remark 2.0.3.** For this  $(n, t)$ -secret sharing scheme it is easy to generate a new secret without changing the shares from the participants. If the subspace  $V \subset W$  is fixed it is possible to calculate for each element  $v$  in  $V$  the element  $v^*$ , such that  $v$  is the closest element in  $V$  to  $v^*$ . This can be done as explained in step 5. of the dealer.

**Remark 2.0.4.** If a CFRZ-scheme, which is a  $(n, t)$ -secret sharing scheme, was executed and the secret  $v$  is known to all participants, then the shares of the participants can be used to realize a CFRZ-scheme, which is a  $(n, t - 1)$ -secret sharing scheme. Therefore, a new secret  $v_{new} \in V$  and the corresponding element  $v_{new}^* \in W \setminus V$ , with  $v_{new}$  the closest vector in  $V$  to the vector  $v_{new}^*$ , are needed. The dealer sends either  $v_{new}^*$  to each participant or publishes this element. The known secret  $v$  is an element in  $V$  and hence it is possible to replace a share of one participant by  $v$  and therefore, with the knowledge of  $v$ , it is possible that the  $(n, t)$ -secret sharing scheme is reduced to a  $(n, t - 1)$ -secret sharing scheme.

## 2.1. Inner product spaces and a private key cryptosystem (Protocol 1)

Now, we extend the  $(n, t)$ -secret sharing scheme from above to **Protocol 1**, a private key cryptosystem, using Remark 2.0.3. Suppose Bob wants to send a message to Alice.

We agree analogously as in the  $(n, t)$ -secret sharing protocol above on the following notations: The closest vector in  $V$  to  $v^* \in W \setminus V$  is denoted with  $v$ . The dimension of  $W$  is denoted by  $m$  and the dimension of  $V$  is denoted by  $t$ , it is  $t < m$ . The set  $B := \{b_1, b_2, \dots, b_t\}$  denotes a basis for the subspace  $V$ .

Private keys:

First Alice and Bob agree on a private key which consists of a subspace  $V$  of a real inner product space  $W$  with dimension  $m$ , it is  $V \subset W$  with  $\dim(V) = t$  and  $t < m$ . The private key is a basis for this subspace  $V$ . For encryption Bob needs an arbitrary basis of the subspace  $V$  and the orthogonal complement  $V^\perp$  to  $V$ . For decryption Alice needs an orthonormal basis of the subspace  $V$ . As soon as they agree on a subspace  $V$  Alice is able to calculate an orthonormal basis for  $V$  and saves this as her private decryption key. As soon as Bob knows  $V$  he calculates a basis  $B^\perp$  for the orthogonal complement  $V^\perp$  to  $V$  which is part of his encryption key.

**Remark 2.1.1.** Let the plaintext  $p$  be an element in the real inner product space  $W$ . Now, we get two possibilities relating to  $p$  and the subspace  $V \subset W$ .

1.  $p \notin V$ : Bob needs an additional vector  $w \in V$ . Therefore he calculates the vector  $w^*$ , which he sends as encrypted message. To receive the original text (the plaintext  $p$ ), the vector  $w^*$  is sent with the vector  $v := w - p$ . The ciphertext, which is sent to Alice, is now the tuple  $c := (w^*, v)$ .
2.  $p \in V$ : Now, the plaintext is an element in the subspace  $V$ . The encrypted message is  $c := p^*$ .

Now, the sent ciphertext is on one hand an element in  $W$ , case 2., and on the other hand a tuple of elements in  $W$ , case 1. In order that no adversary can obtain additional information by looking at the form of  $c$ , Bob must act for each encryption as in case 1. explained. Hence, in both cases he has to fulfill the same steps.

Encryption procedure for Bob:

Let  $p \in W$  be Bob's plaintext.

1. Bob chooses an arbitrary ephemeral vector  $w \in V$ , with  $w \neq p$ , and calculates the vector  $v := w - p$ .
2. In the next step he computes the vector  $w^* \in W \setminus V$  as follows:
  - a) Bob generates the orthogonal complement

$$V^\perp = \{v' \in W \mid v' \perp u \text{ for all } u \in V\}$$

to the subspace  $V$ .

- b) Let  $B^\perp = \{u_1^\perp, u_2^\perp, \dots, u_{m-t}^\perp\}$  be a basis for the orthogonal complement  $V^\perp$ . Now, Bob can compute  $w^*$ :

$$w^* = \underbrace{w}_{\in V} + \underbrace{(\alpha_1 u_1^\perp + \alpha_2 u_2^\perp + \dots + \alpha_{m-t} u_{m-t}^\perp)}_{=: w^\perp \in V^\perp} \in W \setminus V,$$

with  $\alpha_i \in \mathbb{R}$  and at least one  $\alpha_i \neq 0$ , with  $1 \leq i \leq m - t$ .

3. The ciphertext is  $c := (w^*, v)$ . Bob transmits  $c$  to Alice.

Decryption procedure for Alice:

For the decryption Alice needs an orthonormal basis for the subspace  $V$ . She can use the Gram-Schmidt procedure and normalization (see for instance [Atk89]). Assume she gets the orthonormal basis  $G = \{e_1, e_2, \dots, e_t\}$  for the subspace  $V$ .

To get the plaintext, Alice first calculates  $w$ :

$$w = \langle w^*, e_1 \rangle e_1 + \langle w^*, e_2 \rangle e_2 + \dots + \langle w^*, e_t \rangle e_t. \quad (2.1)$$

Then she calculates the plaintext:

$$w - v = p.$$

The **Protocol 1** is summarized in Table 2.1 (page 65).



Table 2.1.: Summary of **Protocol 1**: CV-private key cryptosystem

<b>Private Parameters</b>	
A subspace $V \subset W$ with $\dim(V) = t < m$ of a real inner product space $W$ with $\dim(W) = m$ .	
Alice	Bob
Key Creation	
Calculate an orthonormal basis  $G = \{e_1, e_2, \dots, e_t\}$  for $V$ .	Calculate the orthogonal complement $V^\perp$ to $V$ and a basis  $B^\perp = \{u_1^\perp, u_2^\perp, \dots, u_{m-t}^\perp\}$  for $V^\perp$ .
Encryption	
	Choose plaintext $p \in W$ . Choose arbitrary ephemeral vector $w \in V$ , with $w \neq p$ , and calculate $v := w - p$ . Compute $w^* \in W \setminus V$ :  $w^* = \underbrace{w}_{\in V} + \underbrace{(\alpha_1 u_1^\perp + \alpha_2 u_2^\perp + \dots + \alpha_{m-t} u_{m-t}^\perp)}_{=: w^\perp \in V^\perp},$ $\alpha_i \in \mathbb{R}$ and at least one $\alpha_i \neq 0$ , $1 \leq i \leq m-t$ . Send $c := (w^*, v)$ to Alice.  $c := (w^*, v)$
←	
Decryption	
Compute  $w = \langle w^*, e_1 \rangle e_1 + \langle w^*, e_2 \rangle e_2 + \dots + \langle w^*, e_t \rangle e_t$  and the message is $w - v = p$ .	

**Remark 2.1.2.** The ephemeral key (vector)  $w \in V$  is used only once. Assume that  $(w_1^*, v_1)$  and  $(w_2^*, v_2)$  are two ciphertexts for different plaintexts  $p_1$  and  $p_2$ , respectively, but  $w_1 = w_2$ . If Eve gets  $(w_1^*, v_1)$  and the corresponding plaintext  $p_1$  she is able to calculate  $w_1$ , because  $w_1 = p_1 + v_1$ . Thus, she is able to generate  $p_2$  since  $p_2 = w_2 - v_2 = w_1 - v_2$ , recall we assume  $w_2 = w_1$  and  $w_1$  is known by her. Therefore, the ephemeral key (vector)  $w \in V$  is used only once.

**Remark 2.1.3.** Alice's private key is an orthonormal basis  $G$  for the subspace  $V$ . Hence, if she gets the message  $c := (w^*, v)$  from Bob she only needs to execute formula (2.1) to reconstruct  $w$ . For this she needs  $t$  scalar products. Each scalar product needs  $m$  multiplications and  $m - 1$  additions. Furthermore she needs  $t$  multiplications and  $t - 1$  additions. Altogether these are  $t(m + (m - 1)) + t + (t - 1)$  operations and hence a running time of  $\mathcal{O}(tm + t)$ . Whereby  $m$  is the dimension of the real inner product space  $W$  and  $t$  is the dimension of the subspace  $V \subset W$ . To get the message  $p$  her final step is the subtraction  $w - v$ . For this she needs  $m$  subtractions. Thus, the total running time for decryption is  $\mathcal{O}(tm + t)$ .

**Security 2.1.4.** If the same plaintext  $p \in W$  is encrypted twice we get different ciphertexts. For encryption Bob chooses randomly ephemeral elements  $w \in V$ , thus for the first decryption

of  $p$  he chooses the element  $w_1 \in V$  and for the second he chooses  $w_2 \in V$  with  $w_2 \neq w_1$  and therefore  $v_1 := w_1 - p \neq w_2 - p =: v_2$  and hence  $c_1 = (w_1^*, v_1)$  is different to  $c_2 = (w_2^*, v_2)$  even if they encrypt the same plaintext  $p$ . Thus, for the ciphertext, a statistical frequency attack (see for instance [BFKR15]) is useless.

If an eavesdropper, Eve, gets only ciphertexts (**known ciphertext attack**), that is, she gets a system of tuples  $c_i = (w_i^*, v_i)$ , she knows that  $w_i^* \in W \setminus V$  and for the plaintext  $p_i$  it is  $p_i = w_i - v_i$ . If she does not know the subspace  $V$  she is not able to calculate  $w_i$  from  $w_i^*$ . From the knowledge of  $w_i^*$  she cannot get the subspace  $V$ , because there are infinitely many subspaces of dimension  $V$ , which do not contain the element  $w_i^* \in W \setminus V$  and are candidates for  $V$ , especially if  $\dim(V) \ll \dim(W)$ .

If Eve is able to get ciphertexts and the corresponding plaintexts (for example by a **chosen ciphertext attack**) she can calculate elements of the used subspace  $V$ . Assume that she knows  $c = (w^*, v)$  and the corresponding plaintext  $p$  then she gets the element  $w \in V$  by calculating  $p + v = w$  and hence she knows an element of  $V$ . In the worst case scenario she gets  $t$  ciphertexts and corresponding plaintexts and hence she is able to calculate  $t$  elements of  $V$ . If these  $t$  elements form a basis for  $V$ , she is able to encrypt each message which is further send by Alice and Bob using  $V$ .

In a **chosen plaintext attack** Eve gets to chosen plaintexts the corresponding ciphertexts and hence also pairs of  $c = (w^*, v)$  with the corresponding plaintext  $p$ . With these pairs she is also able, as above, to calculate elements  $w$  in  $V$ , with  $p + v = w$ . If she gets  $t$  independent different elements of  $V$  she gets a basis of  $V$  and hence she is able to encrypt each message which is further send by Alice and Bob using  $V$ .

Therefore, Alice and Bob should change the subspace  $V$  of dimension  $t$  after transmitting  $t$  ciphertexts to avoid chosen plaintext and chosen ciphertext attacks.

**Example 2.1.5.** The calculations in Maple 16 for this example are given in Appendix C.1.

We assume that Bob wants to send a message to Alice. Therefore, Alice and Bob agree privately on the real inner product space  $\mathbb{R}^6$  and a subspace  $V$  of dimension 3, which is given by the basis elements

$$b_1 := \begin{pmatrix} 66 \\ 20 \\ -34 \\ -21 \\ -50 \\ -79 \end{pmatrix}, b_2 := \begin{pmatrix} -36 \\ -7 \\ -62 \\ -56 \\ 30 \\ -71 \end{pmatrix} \text{ and } b_3 := \begin{pmatrix} -41 \\ 16 \\ -90 \\ -8 \\ 62 \\ 28 \end{pmatrix}.$$

For encryption Bob calculates also a basis  $B^\perp$  for the orthogonal complement  $V^\perp$  to  $V$ . This is

$$B^\perp := (u_1^\perp, u_2^\perp, u_3^\perp) := \left( \begin{pmatrix} \frac{67528}{61217} \\ -72236 \\ \frac{61217}{61217} \\ -1433 \\ \frac{61217}{61217} \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{129349}{61217} \\ -362352 \\ \frac{61217}{61217} \\ -208597 \\ \frac{122434}{61217} \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \frac{46287}{61217} \\ -191417 \\ \frac{61217}{61217} \\ -121115 \\ \frac{122434}{61217} \\ 1 \\ 0 \\ 0 \end{pmatrix} \right).$$

With the knowledge of  $b_1, b_2$  and  $b_3$  Alice is able to calculate her decryption key, which is an orthonormal basis  $G = \{e_1, e_2, e_3\}$  for the subspace  $V$ . For this she uses the Gram-Schmidt

procedure and a normalization (see for instance [Atk89]). Thus, she gets

$$e_1 := \begin{pmatrix} \frac{33\sqrt{15094}}{7547} \\ \frac{10\sqrt{15094}}{7547} \\ \frac{17\sqrt{15094}}{7547} \\ \frac{21\sqrt{15094}}{15094} \\ \frac{25\sqrt{15094}}{7547} \\ \frac{79\sqrt{15094}}{15094} \end{pmatrix}, e_2 := \begin{pmatrix} \frac{144211\sqrt{12849748610}}{32124371525} \\ \frac{101599\sqrt{12849748610}}{96373114575} \\ \frac{25667\sqrt{12849748610}}{6424874305} \\ \frac{742847\sqrt{12849748610}}{192746229150} \\ \frac{69667\sqrt{12849748610}}{19274622915} \\ \frac{228797\sqrt{12849748610}}{64248743050} \end{pmatrix}$$

and

$$e_3 := \begin{pmatrix} \frac{1141824503\sqrt{1211483474194866265}}{6057417370974331325} \\ \frac{5968746002\sqrt{1211483474194866265}}{18172252112922993975} \\ \frac{830820734\sqrt{1211483474194866265}}{1211483474194866265} \\ \frac{3745334828\sqrt{1211483474194866265}}{18172252112922993975} \\ \frac{594555334\sqrt{1211483474194866265}}{3634450422584598795} \\ \frac{2286429878\sqrt{1211483474194866265}}{6057417370974331325} \end{pmatrix}.$$

Alice decryption key is the orthonormal basis  $G = \{e_1, e_2, e_3\}$ .

Next, we take a closer look at the encryption procedure. Bob's plaintext is  $p = \begin{pmatrix} 3 \\ 18 \\ 25 \\ 16 \\ 20 \\ 15 \end{pmatrix}$ . He

chooses  $w \in V$  as  $w = 3b_1 - 6b_2 + 7b_3 = \begin{pmatrix} 127 \\ 214 \\ -360 \\ 217 \\ 104 \\ 385 \end{pmatrix}$ , hence it is  $v = w - p = \begin{pmatrix} 124 \\ 196 \\ -385 \\ 201 \\ 84 \\ 370 \end{pmatrix}$ . In

addition he calculates  $w^*$  as

$$w^* = w + (13u_1^\perp - 65u_2^\perp + 5u_3^\perp) = \begin{pmatrix} \frac{476173}{61217} \\ \frac{34757165}{61217} \\ \frac{-15580134}{61217} \\ 222 \\ 117 \\ 320 \end{pmatrix}.$$

He sends the ciphertext  $(w^*, v)$  to Alice.

For decryption, Alice uses her decryption key  $G = \{e_1, e_2, e_3\}$  and the Closest Vector Theorem to calculate

$$U = \langle w^*, e_1 \rangle e_1 + \langle w^*, e_2 \rangle e_2 + \langle w^*, e_3 \rangle e_3$$

and get  $P = U - v = \begin{pmatrix} 3 \\ 18 \\ 25 \\ 16 \\ 20 \\ 15 \end{pmatrix}$ , which is exactly the plaintext  $p$  from Bob.

## 2.2. Inner product spaces and a challenge and response protocol (Protocol 2)

We use the idea behind the CFRZ-scheme from the paper [FMR13] to develop **Protocol 2**, the CV-challenge and response protocol. More precisely this is a symmetric key authentication protocol (see for example [BBFT10] or [BNS10, Section 18.3]).

First we start with a general outline of this challenge and response system. The structure is adapted on a model which is now used for most password and password back-up schemes, see [BBFT10, p. 6]. Afterwards we make suggestions for possible challenges and give a security analysis.

General outline of this symmetric key authentication protocol:

In this variation each prover is assigned to a subspace  $V$  of a real inner product space  $W$ . Due to this, the common shared secret between the prover and the verifier is the tuple  $(P, V)$  where  $P$  is a standard password for the prover and  $V$  is the associated challenge space.

This is a symmetric key authentication protocol, thus, both the prover and verifier use a single common private key within the authentication process, which is here  $V$ .

1. The prover and verifier communicate directly, either face-to-face or by a public key method, to setup a common shared secret  $(P, V)$ , with  $P$  a standard password and  $V$  the challenge subspace of a real inner product space  $W$ . As above, in the previous section, it is  $t := \dim(V) < \dim(W) =: m$ . Each prover's challenge subspace is unique to that

prover. The password is chosen by the prover while the challenge subspace is randomly chosen.

2. The prover presents the password to the verifier. The verifier presents a “question” (see possible challenges for the prover below). The assumption is that this “question” is difficult in the sense that it is infeasible to answer if the subspace  $V$  is unknown. This is repeated a finite number of times. If all answers are correct the prover (and the password) is verified.
3. The cryptographic protocol is then repeated from the viewpoint of the prover, authenticating the verifier to the prover.

We give examples for questions which are very unlikely to answer correctly if the challenge subspace is unknown.

Possible challenges for the prover:

1. How long is the “line” between  $\ell \geq 3$  associated vectors  $v_1, v_2, \dots, v_\ell \in V$  given the vectors  $v_1^*, v_2^*, \dots, v_\ell^* \in W \setminus V$ ? That means, calculate

$$R := \sum_{i=1}^{\ell-1} \|v_i - v_{i+1}\| + \|v_1 - v_\ell\|,$$

whereby  $\|\cdot\|$  denotes the euclidean norm in  $W$ . The prover sends  $R$  as response to the verifier.

In general the “line” between the elements  $v_1, v_2, \dots, v_\ell \in V$  is of a different length than the “line” between the associated vectors  $v_1^*, v_2^*, \dots, v_\ell^* \in W \setminus V$ . The verifier can determine the computer accuracy for the response, for example he can ask for 12 digits of the “length”. Such a situation is visualized in Figure 2.2, that is, given the elements  $v_1^*, v_2^*$  and  $v_3^*$  it is asked after the length of the blue dotted line.

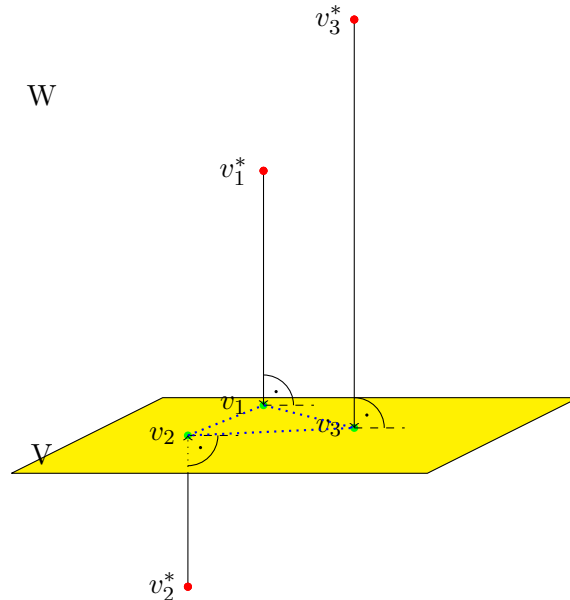


Figure 2.2.: Visualization of a situation in a challenge and response system with  $W = \mathbb{R}^3$  and  $V$  a two dimensional subspace, visualized by a yellow area

2. What is the sum of the entries of the associated vector  $v \in V$  given  $v^* \in W \setminus V$ ? That means the verifier sends  $v^*$  to the prover, the prover calculates  $v = (v_1, v_2, \dots, v_t) \in V$  with the help of the Closest Vector Theorem and

$$R := \sum_{i=1}^t v_i.$$

The result  $R$  is the response from the prover for the verifier.

The **Protocol 2** is summarized in Table 2.2 (page 71).

**Security 2.2.1.** For security analysis we assume that an adversary or eavesdropper has access to the encrypted form of the transmission but is passive in that the adversary will not change any transmissions.

An eavesdropper, Eve, gets elements  $v_1^*, v_2^*, \dots, v_\ell^* \in W \setminus V$ , as mentioned above (see Security 2.1.4) she cannot get the subspace  $V$ , because there are infinitely many subspaces of dimension  $V$ , which do not contain the element  $w_i^* \in W \setminus V$ ,  $1 \leq i \leq \ell$ , and are candidates for  $V$ , especially if  $\dim(V) \ll \dim(W)$ . As response she gets a number which is the length of the “line” between the associated elements  $v_1, v_2, \dots, v_\ell \in V$  to  $v_1^*, v_2^*, \dots, v_\ell^* \in W \setminus V$ . This gives also no hint for the subspace  $V$ . In the second variation for the challenges it is asks after the sum of entries of a vector  $v \in V$  associated to  $v^* \in W \setminus V$ . The response gives not enough information for  $v$  and for the subspace  $V$  if only  $v^* \in W \setminus V$  is known.

There are infinitely many numbers of possible challenges of kind 1. and 2., see above, thus no challenge is used twice by the verifier. Therefore, replay attacks, in which an adversary records a communication session and replays parts of the session or the whole session ([MvOV97]), is avoided.

**Example 2.2.2.** An example for one challenge and the corresponding response together with the Maple-Code can be found in Appendix C.2.

It is possible to get a two-way authentication with this challenge and response system. That means the prover authenticates the verifier in the time where the verifier authenticates the prover. For this the prover and verifier start as above and agree on the common shared secret  $(P, V)$  with  $V$  the challenge space of a real inner product space  $W$  also with  $t := \dim(V) < \dim(W) =: m$ . Before the verifier gives challenges to the prover the prover transmits a distance to the verifier in which the length of the “line” must be (or respectively the sum of the entries of an element in  $V$  if we are in case 2. for the challenges), that means  $L_1$  and  $L_2 \in \mathbb{R}$  are given to the verifier. The verifier chooses  $v_i \in V$ , such that the expected response from the prover is between  $R_1$  and  $R_2$  and sends the corresponding elements  $v_i^* \in W \setminus V$  to the prover. The prover calculates the exact response  $R$  (up to a given computational accuracy) and sends it to the verifier. It is very unlikely that a person is able to give elements  $v_i^*$  as challenges to the prover, such that the calculated response  $R$  lies in the expected range, that is,  $L_1 \leq R \leq L_2$ .

Table 2.2.: Summary of **Protocol 2**: CV-challenge and response protocol

<b>Private Parameters</b>	
Subspace $V \subset W$ , with $\dim(V) = t < m$ , of a real inner product space $W$ , with $\dim(W) = m$ , and a common password $P$ . The shared secret is $(P, V)$ .	
Verifier	Prover
Calculate orthogonal complement $V^\perp$ to $V$ and a basis $B^\perp = \{u_1^\perp, u_2^\perp, \dots, u_{m-t}^\perp\}$ for $V^\perp$ .	Calculate orthonormal basis $G = \{e_1, e_2, \dots, e_t\}$ for $V$ .
$\longleftarrow$ Present the password $P$ to the verifier $P$	
Take challenge space $V$ corresponding to password $P$ , more precisely the calculated orthogonal basis $B^\perp$ . Choose elements $v_1, v_2, \dots, v_\ell \in V$ with $\ell \geq 3$ and calculate the associated elements $v_1^*, v_2^*, \dots, v_\ell^* \in W \setminus V$ . Compute $v_i^* \in W \setminus V$ , $1 \leq i \leq \ell$ : $v_i^* = \underbrace{v_i}_{\in V} + \underbrace{(\alpha_{i_1} u_1^\perp + \alpha_{i_2} u_2^\perp + \dots + \alpha_{i_{m-t}} u_{m-t}^\perp)}_{=: v^\perp \in V^\perp},$ $\alpha_{i_j} \in \mathbb{R}$ and at least one $\alpha_{i_j} \neq 0$ , $1 \leq j \leq m-t$ . Send $v_1^*, v_2^*, \dots, v_\ell^*$ as challenge to the prover.	$\xrightarrow{\text{Challenge: } v_1^*, v_2^*, \dots, v_\ell^*}$
Calculate $R' := \sum_{i=1}^{\ell-1} \ v_i - v_{i+1}\  + \ v_1 - v_\ell\ .$	Compute $v_i = \langle v_i^*, e_1 \rangle e_1 + \langle v_i^*, e_2 \rangle e_2 + \dots + \langle v_i^*, e_t \rangle e_t$ for each $v_i^*$ , $1 \leq i \leq \ell$ . Calculate the response $R$ and send it to the verifier, it is $R := \sum_{i=1}^{\ell-1} \ v_i - v_{i+1}\  + \ v_1 - v_\ell\ .$
$\longleftarrow$ Response: $R$	
Proof if $R' = R$ .	





## Chapter 3

# A group theoretical ElGamal cryptosystem based on a semidirect product of groups

In this chapter **Protocol 3**, a group theoretical ElGamal cryptosystem, is introduced, which is based on semidirect products of groups. It extends the key exchange protocol based on a semidirect product of (semi)groups introduced in [HKKS13] by M. Habeeb, D. Kahrobaei, C. Koupparis and V. Shpilrain, also called HKKS-key exchange protocol or shorter HKKS-scheme. We also explain a proposal for a signature protocol, which is **Protocol 4**. These cryptographic protocols are published in [Mol15].

There is an ongoing research about the HKKS-scheme with linear algebra attacks and researches about suitable platforms which also affects the cryptosystem (**Protocol 3**) and the signature protocol (**Protocol 4**). An overview of this research will be given.

We start with the definition of a semidirect product and a recall of the HKKS-scheme. It follows the introduction of **Protocol 3**, **Protocol 4** and an overview about ongoing research of the HKKS-scheme closes this chapter. For this chapter we are orientated on [HKKS13] and [Mol15].

**Definition 3.0.1.** [Rot95]

Let  $G$  and  $H$  be two groups, let  $Aut(G)$  be the group of automorphisms of  $G$  and let  $\rho : H \rightarrow Aut(G)$  be a homomorphism. Then the semidirect product of  $G$  and  $H$  is the set

$$\Gamma = G \rtimes_{\rho} H = \{(g, h) \mid g \in G, h \in H\}$$

with the group operation given by

$$(g, h) \cdot (g', h') = (g^{\rho(h')}, g', h \cdot h').$$

Here  $g^{\rho(h')}$  denotes the image of  $g$  under the automorphism  $\rho(h')$ .

One special case of the semidirect product construction is where the group  $H$  is a subgroup of the group  $Aut(G)$ . If  $H = Aut(G)$ , then the corresponding semidirect product is called the **holomorph** of the group  $G$ . Thus, the holomorph of  $G$ , usually denoted by  $Hol(G)$ , is the set

$$Hol(G) = \{(g, \phi) \mid g \in G, \phi \in Aut(G)\}$$

with the group operation given by

$$(g, \phi) \cdot (g', \phi') = (\phi'(g), g', \phi \cdot \phi').$$

A product  $\phi \cdot \phi'$  of two homomorphisms means that  $\phi$  is applied first. It is often more practical to use a subgroup of  $Aut(G)$  in this construction, as it is done in [HKKS13, Section 3], where a key exchange protocol is described, that uses (as the platform) an extension of a group  $G$  by a cyclic group of automorphisms. This key exchange is described in more details below.

**Remark 3.0.2.** This construction can also be used if  $G$  is not necessarily a group, but just a semigroup, and/or if endomorphisms of  $G$ , that are not necessarily automorphisms of  $G$ , are considered. Then the result will be a semigroup.

Now, we describe the key exchange protocol based on a semidirect product of (semi)groups by automorphisms from M. Habeeb, D. Kahrobaei, C. Koupparis and V. Shpilrain (see [HKKS13]).

Alice and Bob use a group (or semigroup)  $G$  and they can use just a cyclic subgroup  $H$  (or a cyclic subsemigroup) of the group  $Aut(G)$  (respectively, of the semigroup  $End(G)$  of endomorphisms) instead of the whole group of automorphisms of  $G$ .

Let  $G$  be a (semi)group. An element  $g \in G$  as well as an arbitrary automorphism  $\phi \in Aut(G)$  (or an arbitrary endomorphism  $\phi \in End(G)$ ) are chosen and published.

Both, Alice and Bob, are going to work with elements of the form  $(g, \phi^r)$ , where  $g \in G$  and  $r \in \mathbb{N}$ . Note that two elements of this form are multiplied as follows:

$$(g, \phi^r) \cdot (h, \phi^s) = (\phi^s(g) \cdot h, \phi^{r+s}).$$

1. Alice chooses a private number  $n \in \mathbb{N}$ ;  
she computes  $(g, \phi)^n = (\phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g, \phi^n)$  and sends only the first component, namely  $a := \phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g$ , to Bob.
2. Bob chooses a private number  $k \in \mathbb{N}$ ;  
he computes  $(g, \phi)^k = (\phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g, \phi^k)$  and sends only the first component, namely  $b := \phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g$ , to Alice.
3. Alice computes  $(b, x) \cdot (a, \phi^n) = (\phi^n(b) \cdot a, x \cdot \phi^n)$ .  
Her key is now  $K_A := \phi^n(b) \cdot a$ . Note that she does not actually “compute”  $x \cdot \phi^n$  because she does not know the automorphism  $x = \phi^k$ ; recall that it was not transmitted to her, but she does not need it to compute  $K_A$ .
4. Bob computes  $(a, y) \cdot (b, \phi^k) = (\phi^k(a) \cdot b, y \cdot \phi^k)$ .  
His key is now  $K_B := \phi^k(a) \cdot b$ . Again, Bob does not actually “compute”  $y \cdot \phi^k$  because he does not know the automorphism  $y = \phi^n$ .
5. Since  $(b, x) \cdot (a, \phi^n) = (a, y) \cdot (b, \phi^k) = (g, \phi)^{n+k}$ , it is  $K_A = K_B = K$ , the shared secret key.

**Remark 3.0.3.** The shared secret key is  $K = K_B = K_A$ , since

$$\begin{aligned} K_B &= \phi^k(a) \cdot b \\ &= \phi^k(\phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g) \cdot \phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g \\ &= \phi^{k+n-1}(g) \cdot \phi^{k+n-2}(g) \cdots \phi^{k+1}(g) \cdot \phi^k(g) \cdot \phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g \\ &= \phi^n(\phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g) \cdot \phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g \\ &= \phi^n(b) \cdot a \\ &= K_A. \end{aligned}$$

**Remark 3.0.4.** In contrast to the standard Diffie-Hellman key exchange (see Section 1.2.1), the correctness here is based on the equality  $h^n \cdot h^k = h^k \cdot h^n = h^{n+k}$  rather than on the equality  $(h^n)^k = (h^k)^n = h^{nk}$ . In the standard Diffie-Hellman set up, the trick would not work, because, if the shared key  $K$  was just the product of two openly transmitted elements, then anybody, including the eavesdropper, could compute  $K$ .

**Remark 3.0.5.** The transmitted elements are products of  $n$  or  $k$ , respectively, elements of a (semi)group  $G$ . To compute powers of an element, Alice and Bob can use the “square-and-multiply” method (see for instance [HPS08]), as it is also done in the standard Diffie-Hellman key exchange. The cost of applying an automorphism  $\phi$  to an element  $g \in G$ , and also of computing powers of  $\phi$  depends on the platform (semi)group  $G$  and automorphism  $\phi$ , which are used.

Let  $p$  be a prime number. We denote the field with  $p$  elements with  $\mathbb{F}_p$  and  $\mathbb{F}_p^*$  denotes the multiplicative subgroup. Here  $Mat(3, \mathbb{F}_7[A_5])$  means the set of  $3 \times 3$  matrices with entries in  $\mathbb{F}_7[A_5]$ , whereby  $\mathbb{F}_7$  is the field with seven elements and  $A_5$  is the group of even permutations on five symbols.

For  $G = \mathbb{F}_p^*$  (here more precisely  $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^*$ ) with  $\phi(h) = h^\ell$  (as in Example 3.1.2), and  $G = Mat(3, \mathbb{F}_7[A_5])$  with  $\phi_H(L) = H^{-1}LH$  (as for the signature in Section 3.2) M. Habbeb, D. Kahrobaei, C. Koupparis and V. Shpilrain mention in [HKKS13] that the cost of computing  $(g, \phi)^n$  is  $\mathcal{O}(\log n)$ , just as in the standard Diffie-Hellman protocol.

### 3.1. ElGamal like public key cryptosystem (Protocol 3)

**Protocol 3**, a public key cryptosystem, is an ElGamal like cryptosystem and it is based on a semidirect product of groups. After the general description of this cryptosystem we give two examples for possible platform groups and discuss their security. For this we are orientated on [Mol15].

Alice and Bob can use a group  $G$  and a cyclic subgroup  $H$  of the group  $Aut(G)$  instead of the whole group of automorphisms of  $G$  as in the key exchange protocol.

1. Alice and Bob agree on an element  $g \in G$  and an automorphism  $\phi \in H \subseteq Aut(G)$  or Alice determines  $g$  and  $\phi$  as public parameters, respectively. Whereby, Alice has to take care, that the base element  $(g, \phi)$  has a large order, otherwise the system is susceptible to brute force attacks.
2. Alice chooses a random natural number  $n$  as her secret key. She computes  $(g, \phi)^n = (\phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g, \phi^n)$  and publishes the first component  $a := \phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g$  only.
3. Bob wants to send the message  $m \in G$  to Alice. He picks a random ephemeral key  $k \in \mathbb{N}$ . Therefore, he has to calculate two elements. He computes  $(g, \phi)^k = (\phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g, \phi^k)$ , its first component is named  $c_1 := \phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g$ . Then he computes  $(a, y) \cdot (c_1, \phi^k) = (\phi^k(a) \cdot c_1, y \cdot \phi^k)$ . He sets the first component  $b := \phi^k(a) \cdot c_1$ . Note that he does not actually “compute”  $y \cdot \phi^k$ , because he does not know the automorphism  $y = \phi^n$ , but he does not need it to compute  $b$ . He computes  $c_2 := b \cdot m = \phi^k(a) \cdot c_1 \cdot m$  and sends the ciphertext  $(c_1, c_2)$  to Alice.
4. Alice computes  $(c_1, x) \cdot (a, \phi^n) = (\phi^n(c_1) \cdot a, x \cdot \phi^n)$ , named the first component  $K := \phi^n(c_1) \cdot a$  and recovers the message by  $m = K^{-1} \cdot c_2 = (\phi^n(c_1) \cdot a)^{-1} \cdot c_2$ . Note that she does not “compute”  $x \cdot \phi^n$  because she does not know  $x = \phi^k$  and does not need it to compute  $K$ .

Alice gets the message  $m$ , because from

$$K^{-1} \cdot c_2 = (\phi^n(c_1) \cdot a)^{-1} \cdot c_2 = (\phi^n(c_1) \cdot a)^{-1} \cdot \phi^k(a) \cdot c_1 \cdot m$$

with

$$\phi^n(c_1) \cdot a = \phi^k(a) \cdot c_1,$$

which follows from the same calculations as in Remark 3.0.3, it is

$$\begin{aligned} K^{-1} \cdot c_2 &= (\phi^n(c_1) \cdot a)^{-1} \cdot c_2 \\ &= (\phi^n(c_1) \cdot a)^{-1} \cdot \phi^k(a) \cdot c_1 \cdot m \\ &= (\phi^n(c_1) \cdot a)^{-1} \cdot \phi^n(c_1) \cdot a \cdot m \\ &= m. \end{aligned}$$

**Protocol 3**, an ElGamal like cryptosystem, is summarized in Table 3.1 (page 76).

Table 3.1.: Summary of **Protocol 3**: Group theoretical ElGamal like public key cryptosystem using semidirect products

<b>Public Parameters</b>	
Group $G$ and cyclic subgroup $H$ of the group $Aut(G)$ , $g \in G$ and $\phi \in H \subseteq Aut(G)$ .	
<b>Alice</b>	<b>Bob</b>
<b>Key Creation</b>	
Choose private key $n \in \mathbb{N}$ . Compute $(a, \phi^n) := (g, \phi)^n$ with $a := \phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g$ . Publish $a$ .	
<b>Encryption</b>	
	Choose plaintext $m \in G$ . Choose random ephemeral key $k \in \mathbb{N}$ . Compute $(c_1, \phi^k) := (g, \phi)^k$ with $c_1 := \phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g$ , $(a, y) \cdot (c_1, \phi^k) = \underbrace{(\phi^k(a) \cdot c_1)}_{=:b} \cdot y \cdot \phi^k$  and $c_2 := b \cdot m = \phi^k(a) \cdot c_1 \cdot m$ . Send ciphertext $(c_1, c_2)$ to Alice.  $\xleftarrow{(c_1, c_2)}$
<b>Decryption</b>	
Compute $(c_1, x) \cdot (a, \phi^n) = \underbrace{(\phi^n(c_1) \cdot a)}_{=:K} \cdot x \cdot \phi^n$  and recover $m = K^{-1} \cdot c_2$ .	

**Remark 3.1.1.** Alice computes a large power of the element  $(g, \phi)$ , but she does not transmit the whole result, she only publishes the part  $a$  of it. Bob also computes a large power of the element  $(g, \phi)$  and only the first part  $c_1$  is a part of his ciphertext. In addition, he computes a

product of two elements from  $G$  and only the first part multiplied by the message is the second part of his ciphertext.

It is important that different random ephemeral keys  $k$  are used to encrypt different messages. As it is for the standard ElGamal cryptosystem (see [MvOV97]). Suppose that Bob uses the same ephemeral key  $k$  to encrypt two messages  $m_1$  and  $m_2$  and assume that  $m_1$  is known. The ciphertext pairs are  $(c_1, c_2)$  and  $(c'_1, c'_2)$ , with  $c_1 = c'_1$ ,  $c_2 = \phi^k(a) \cdot c_1 \cdot m_1$  and  $c'_2 = \phi^k(a) \cdot c'_1 \cdot m_2$ . Eve only has to calculate  $m_1 \cdot (c_2)^{-1} \cdot c'_2$  to get the message  $m_2$ .

Another non-commutative generalization of the ElGamal key exchange which is based on the complexity differences between various group-theoretic decision problems and uses polycyclic groups can be found in [KK06].

**Example 3.1.2.** Following the example in [HKKS13, Chapter 5] for the key exchange presented there, we now use the multiplicative group  $\mathbb{F}_p^*$  (here more precisely  $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^*$ ) as the platform group  $G$  for illustration purposes.

Let  $G$  be the multiplicative group  $\mathbb{F}_p^*$  with  $p$  a prime number.

For the endomorphism  $\phi$  of the group  $\mathbb{F}_p^*$  a number  $\ell \in \mathbb{N}$ ,  $\ell > 1$ , is selected, such that

$$\phi(h) = h^\ell \quad \text{for every } h \in \mathbb{F}_p^*.$$

If  $\ell$  is relatively prime to  $p - 1$ , then  $\phi$  is actually an automorphism.

For an element  $g \in \mathbb{F}_p^*$  and  $n \in \mathbb{N}$  it is

$$(g, \phi)^n = (\phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g, \phi^n)$$

with

$$\begin{aligned} \phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g &= g^{\ell^{n-1}} \cdot g^{\ell^{n-2}} \cdots g^\ell \cdot g \\ &= g^{\ell^{n-1} + \ell^{n-2} + \cdots + \ell + 1} \\ &= g^{\frac{\ell^n - 1}{\ell - 1}}, \end{aligned}$$

since the finite geometric sum is used and

$$\phi^r(g) = g^{\ell^r} \quad \text{for all } r \in \mathbb{N}.$$

An example is performed in Table 3.2 (page 78).

Table 3.2.: Example with  $G = \mathbb{F}_p^*$ .

<b>Public Parameters</b>	
$G = \mathbb{F}_p^*$ with $p$ a prime number, $\phi(h) = h^\ell \forall h \in \mathbb{F}_p^*$ with qualified $\ell \in \mathbb{N}$ , $\ell > 1$ , and $g \in \mathbb{F}_p^*$ .	
Alice	Bob
Key Creation	
Choose private key $n \in \mathbb{N}$ . Compute $(a, \phi^n) := (g, \phi)^n$ with $a := \phi^{n-1}(g) \cdot \phi^{n-2}(g) \cdots \phi(g) \cdot g = g^{\frac{\ell^n - 1}{\ell - 1}}$ . Publish $a$ .	
Encryption	
	Choose plaintext $m \in \mathbb{F}_p^*$ . Choose random ephemeral key $k \in \mathbb{N}$ . Compute $(c_1, \phi^k) := (g, \phi)^k$ with $c_1 := \phi^{k-1}(g) \cdot \phi^{k-2}(g) \cdots \phi(g) \cdot g$ , $(a, y) \cdot (c_1, \phi^k) = \underbrace{(\phi^k(a) \cdot c_1, y \cdot \phi^k)}_{=:b}$ and $c_2 := b \cdot m = \phi^k(a) \cdot c_1 \cdot m$ . Send $c_1 = g^{\frac{\ell^k - 1}{\ell - 1}}$ and $c_2 = \phi^k(a) \cdot c_1 \cdot m$ $= \phi^k \left( g^{\frac{\ell^n - 1}{\ell - 1}} \right) \cdot g^{\frac{\ell^k - 1}{\ell - 1}} \cdot m$ $= \left( g^{\frac{\ell^n - 1}{\ell - 1}} \right)^{\ell^k} \cdot g^{\frac{\ell^k - 1}{\ell - 1}} \cdot m$ $= g^{\frac{\ell^{k+n} - 1}{\ell - 1}} \cdot m$ as ciphertext $(c_1, c_2)$ to Alice.
$\longleftarrow (c_1, c_2)$	
Decryption	
Compute $(c_1, x) \cdot (a, \phi^n) = \underbrace{(\phi^n(c_1) \cdot a, x \cdot \phi^n)}_{=:K}$ , it is $K = \left( \phi^n \left( g^{\frac{\ell^k - 1}{\ell - 1}} \right) \cdot g^{\frac{\ell^n - 1}{\ell - 1}} \right)$ $= \left( \left( g^{\frac{\ell^k - 1}{\ell - 1}} \right)^{\ell^n} \cdot g^{\frac{\ell^n - 1}{\ell - 1}} \right)$ $= g^{\frac{\ell^{k+n} - 1}{\ell - 1}}.$ Recover $m = K^{-1} \cdot c_2$ $= g^{-\frac{\ell^{k+n} - 1}{\ell - 1}} \cdot g^{\frac{\ell^{k+n} - 1}{\ell - 1}} \cdot m.$	

**Security 3.1.3.** We now take a closer look at the security of the ElGamal like cryptosystem

with the platform group  $G = \mathbb{F}_p^*$ . If the eavesdropper Eve wants to get the message  $m$  by calculation

$$b^{-1} \cdot c_2 = b^{-1} \cdot \underbrace{g^{\frac{\ell^{k+n}-1}{\ell-1}}}_{=b} \cdot m = m$$

she has to know the “key”  $b$ .

On the one hand she can compute  $b$  in two ways by solving the discrete logarithm problem. First she can compute  $b = \phi^n(c_1) \cdot a$ . For this she needs the private key  $n$  from Alice. As an alternative she computes  $b = \phi^k(a) \cdot c_1$ . For this she has to get the ephemeral key  $k$  from Bob. In both ways she has to solve the **discrete logarithm problem** twice. For example, if she wants to get the private ephemeral key  $k$  from Bob she first has to recover  $\frac{\ell^k-1}{\ell-1}$  from  $c_1 := g^{\frac{\ell^k-1}{\ell-1}}$ , and then she has to recover  $k$  from  $\ell^k$ , because  $\ell$  is known since  $\phi$  is published.

On the other hand she can recover  $b$  by the analog of what is called **the Diffie-Hellman problem**, so she should recover  $b := g^{\frac{\ell^{k+n}-1}{\ell-1}}$  from the triple  $\left(g, c_1 := g^{\frac{\ell^k-1}{\ell-1}}, a := g^{\frac{\ell^n-1}{\ell-1}}\right)$ . This is exactly the Diffie-Hellman problem, because Eve knows the elements  $g$  and  $\ell$ , which are public parameters, and it is equivalent to recover  $g^{\ell^{n+k}}$  from the triple  $\left(g, g^{\ell^n}, g^{\ell^k}\right)$ .

Thus, if the group  $G$  is the multiplicative group  $\mathbb{F}_p^*$ , with  $p$  a prime number, then this cryptographic protocol is not really different from the standard ElGamal cryptosystem, described in Section 1.2.2.

Therefore, the standard ElGamal cryptosystem is a special case of this public key cryptosystem, hence, breaking this cryptosystem would imply breaking the ElGamal cryptosystem.

**Example 3.1.4.** We now give an example for the public key cryptosystem with a non-commutative group. Choose a non-commutative group  $G$ , **not** a semigroup, because the inverse of an element  $g^{-k-n}h^{k+n}$ , with  $g, h \in G$ , is needed. For example  $G = \text{GL}(r, \mathbb{K})$ , with  $r \in \mathbb{N}$ ,  $r > 1$ , and a field  $\mathbb{K}$ , the general linear group of  $r \times r$  matrices with entries from a field.

Use an extension of the group  $G$  by an inner automorphism  $\rho_H$  which is conjugated by a matrix  $H \in \text{GL}(r, \mathbb{K})$ . Alice and Bob can use any non-commutative group  $G$  if  $\rho_H$  is selected to be a non-trivial inner automorphism, that is, a conjugation by an element which is not in the center of  $G$ , where the center of  $\text{GL}(r, \mathbb{K})$  is the set defined as

$$C(\text{GL}(r, \mathbb{K})) = \{\alpha \cdot I \mid \alpha \in \mathbb{K} \setminus \{0\} \text{ and } I \text{ the identity matrix in } \text{GL}(r, \mathbb{K})\}.$$

For any Matrix  $M \in G$  and for any  $k \in \mathbb{N}$ ,  $k > 0$ , it is

$$\rho_H(M) = H^{-1}MH \quad \text{and} \quad \rho_H^k(M) = H^{-k}MH^k.$$

For  $s \in \mathbb{N}$ ,  $s > 0$ , it is

$$\begin{aligned} (M, \rho_H)^s &= (H^{-(s-1)}MH^{s-1} \cdot H^{-(s-2)}MH^{s-2} \dots H^{-1}MH \cdot M, \rho_H^s) \\ &= (H^{-s}(HM)^s, \rho_H^s). \end{aligned}$$

An example is performed in Table 3.3 (page 80).

Table 3.3.: Example with  $G = \text{GL}(r, \mathbb{K})$

<b>Public Parameters</b>	
Group $G = \text{GL}(r, \mathbb{K})$ , $r \in \mathbb{N}$ and $r > 1$ , a matrix $H \in G$ , therefore the automorphism is $\rho_H$ , and a matrix $M \in G$ . Take care that $H$ and $HM$ do not commute.	
Alice	Bob
Key Creation	
Choose private key $n \in \mathbb{N}$ . Compute $(a, \rho_H^n) := (M, \rho_H)^n$ with $a := \rho_H^{n-1}(M) \cdot \rho_H^{n-2}(M) \cdots \rho_H(M) \cdot M$ $\quad = H^{-n}(HM)^n$ . Publish $a$ .	
Encryption	
	Choose plaintext $m \in G$ . Choose random ephemeral key $k \in \mathbb{N}$ . Compute $(c_1, \rho_H^k) := (M, \rho_H)^k$ with $c_1 := \rho_H^{k-1}(M) \cdot \rho_H^{k-2}(M) \cdots \rho_H(M) \cdot M$ , $(a, y) \cdot (c_1, \rho_H^k) = \underbrace{(\rho_H^k(a) \cdot c_1, y \cdot \rho_H^k)}_{=:b}$ and $c_2 := b \cdot m = \rho_H^k(a) \cdot c_1 \cdot m$ . Send $c_1 = H^{-k}(HM)^k$ and $c_2 = \rho_H^k(a) \cdot c_1 \cdot m$ $\quad = H^{-k}H^{-n}(HM)^nH^k \cdot H^{-k}(HM)^k \cdot m$ $\quad = H^{-k-n}(HM)^{n+k} \cdot m$ as ciphertext $(c_1, c_2)$ to Alice.
	$\longleftarrow (c_1, c_2)$
Decryption	
Compute $(c_1, x) \cdot (a, \rho_H^n) = \underbrace{(\rho_H^n(c_1) \cdot a, x \cdot \rho_H^n)}_{=:K}$ , it is $K = \rho_H^n(c_1) \cdot a$ $\quad = H^{-n}H^{-k}(HM)^kH^n \cdot H^{-n}(HM)^n$ $\quad = H^{-n-k}(HM)^{k+n}$ . Recover $m = K^{-1} \cdot c_2$ $\quad = (H^{-n-k}(HM)^{k+n})^{-1} \cdot H^{-k-n}(HM)^{n+k} \cdot m$ .	

**Remark 3.1.5.** If the matrices  $H$  and  $HM$  commute, Eve can use  $c_1$  and  $c_2$  to get the element

$$V := c_1^{-1} \cdot c_2 = (HM)^{-k}H^k \cdot H^{-k-n}(HM)^{n+k} \cdot m = H^{-n}(HM)^n \cdot m.$$

The public key is the element  $a = H^{-n}(HM)^n$  and hence everyone could compute  $m$  in the



following way

$$\begin{aligned} a^{-1} \cdot V &= a^{-1} c_1^{-1} \cdot c_2 \\ &= (H^{-n}(HM)^n)^{-1} H^{-n}(HM)^n \cdot m = m. \end{aligned}$$

The inverse of  $a$  and  $c_1$  exists because  $G$  is a group.

To prevent this Alice has to take care that  $H$  and  $HM$  do **not commute**.

**Security 3.1.6.** We look at the security of the public key cryptosystem with the platform group  $G = \text{GL}(r, \mathbb{K})$ ,  $r \in \mathbb{N}$ . As in Security 3.1.3 with the platform group  $G = \mathbb{F}_p^*$  the eavesdropper Eve can get the message  $m$  if she is aware of the “key”  $b$ , it is  $b = H^{-(n+k)}(HM)^{n+k}$ . She then calculates

$$b^{-1} \cdot c_2 = b^{-1} \cdot \underbrace{H^{-(n+k)}(HM)^{n+k}}_{=b} \cdot m = m.$$

For example she can get  $b$  by computing  $b = \rho_H^k(a) \cdot c_1$ . Therefore, she has to try to recover the ephemeral key  $k$  from Bob, that means, she has to recover  $k$  from the element  $c_1 := H^{-k}(HM)^k = g^{-k}h^k$  (with  $g := H$  and  $h := HM$ ). In the special case with  $g = I$  it is the **discrete logarithm problem** for matrices in  $\text{GL}(r, \mathbb{K})$ , which is recover  $k$  from  $h^k$ . It is known (see [MW97]) that a probabilistic polynomial-time reduction of the discrete logarithm problem exists in the general linear group  $\text{GL}(r, q)$  ( $r \times r$  matrices with entries of a finite field with  $q$  elements) to the discrete logarithm problem in some small extension fields of  $\mathbb{F}_q$  (a finite field of order  $q$ , with  $q = p^s$  where  $p$  is the characteristic of  $\mathbb{F}_q$ ). Statistical experiments show that for a random matrix  $M$ , matrices  $M^n$  are indistinguishable from random (see [HKKS13]). Furthermore, the security assumption is that it is computationally hard to reclaim the “key”  $b = H^{-(n+k)}(HM)^{n+k}$  from the quadruple

$$\left( H, M, a := H^{-n}(HM)^n, c_1 := H^{-k}(HM)^k \right).$$

Therefore, Alice has to take care that the matrices  $H$  and  $HM$  do not commute (see Remark 3.1.5).

This example was also given in the work [Mol15] but in the time under review the paper [KMU14] by M. Kreuzer A. D. Myasnikov and A. Ushakov appeared in which they give a linear algebra attack on the HKKS-key exchange protocol with  $G = \text{Mat}(3, \mathbb{F}_7[A_5])$ , which was an example in [HKKS13]. We will explain this in more details in Section 3.3 and also describe a linear decomposition attack by V. Roman’kov.

### 3.2. Signature with a semigroup of $3 \times 3$ matrices over $\mathbb{F}_7[A_5]$ (Protocol 4)

In this section an idea for a signature scheme, **Protocol 4**, inspired by the example of the key exchange protocol with a semigroup as platform group (see [HKKS13, Chapter 6]) is described and a security analysis is given. For this we are orientated on [Mol15].

In [KK12] D. Kahrobaei and C. Koupparis give a survey about several digital signature proposals using non-commutative groups and rings.

Let  $G$  be a non-commutative semigroup which has non-central invertible elements, where  $\rho_H$  is a non-identical inner automorphism, that is, a conjugation by an element  $H \in G$ , such that  $H^{-1}gH \neq g$  for at least some  $g \in G$ .

1. Alice chooses an invertible  $H \in G$  for the automorphism  $\rho_H$  and a qualified hash function  $h$ , with

$$h : \{\text{possible messages}\} \longrightarrow \{\text{non-invertible matrices in } G\}$$

(see Security 3.2.3 (II) and (III)). This is published.

2. Alice picks  $n \in \mathbb{N}$  and an element  $M \in G$  private. She computes  $(M, \rho_H)^n = (\rho_H^{n-1}(M) \cdot \rho_H^{n-2}(M) \cdots \rho_H(M) \cdot M, \rho_H^n)$  and publishes only the first component  $a := \rho_H^{n-1}(M) \cdot \rho_H^{n-2}(M) \cdots \rho_H(M) \cdot M$ . Alice has to take care that  $H$  and  $HM$  do not commute (see Remark 3.2.2) and that her element  $a$  has no inverse in  $G$  (see Security 3.2.3 (I)).
3. To sign the message  $m$  she picks an ephemeral key  $k \in \mathbb{N}$ , and computes  $(M, \rho_H)^k = (\rho_H^{k-1}(M) \cdot \rho_H^{k-2}(M) \cdots \rho_H(M) \cdot M, \rho_H^k)$  with the first component named  $b := \rho_H^{k-1}(M) \cdot \rho_H^{k-2}(M) \cdots \rho_H(M) \cdot M$ . With the help of the hash function  $h$  she computes the element  $Z := h(m) \cdot \rho_H^n(b)$ . Her signature is the quadruple  $(k, b, Z, m)$ .
4. Before Bob can prove the signature he has to calculate the element  $(a, x) \cdot (b, \rho_H^k) = (\rho_H^k(a) \cdot b, x \cdot \rho_H^k)$ . The first component is named  $E := \rho_H^k(a) \cdot b$ . Note that he does not actually “compute”  $x \cdot \rho_H^k$  because he does not know the automorphism  $x = \rho_H^n$ , but he does not need it to compute  $E$ . Bob is aware of the hash function  $h$  and he proves the signature with the calculation  $Z \cdot a = h(m) \cdot E$ .

It is

$$\begin{aligned} Z \cdot a &= h(m) \cdot \rho_H^n(b) \cdot a \\ &= h(m) \cdot \rho_H^k(a) \cdot b \\ &= h(m) \cdot E, \end{aligned}$$

because  $\rho_H^n(b) \cdot a = \rho_H^k(a) \cdot b$ , which follows from the same calculations as in Remark 3.0.3.

Now, let  $G$  be the semigroup of  $3 \times 3$  matrices over the group ring  $\mathbb{F}_7[A_5]$ , where  $A_5$  is the alternating group on 5 elements, that is,  $G = \text{Mat}(3, \mathbb{F}_7[A_5])$ . We can take  $\mathbb{F}_7^* = (\mathbb{Z}/7\mathbb{Z})^*$ . The inner automorphism  $\rho_H$  is a conjugation by a matrix  $H \in \text{GL}(3, \mathbb{F}_7[A_5])$ . It is

$$\rho_H(L) = H^{-1}LH \quad \text{and} \quad \rho_H^r(L) = H^{-r}LH^r,$$

for any matrix  $L \in G$  and any  $r \in \mathbb{N}$ ,  $r > 0$ .

**Remark 3.2.1.** The semigroup of  $3 \times 3$  matrices over the group ring  $\mathbb{F}_7[A_5]$  is used, because the multiplication can be calculated very efficient in this semigroup and it provides a large key space (see [KKS13]).

Note that the element  $a$  has no inverse in  $G$  if  $M$  has no inverse in  $G$ .

A technique to obtain an invertible matrix  $H$  is presented in [HKKS13, Chapter 8]. From there it is also known that the exponents  $n$  and  $k$  should be of the magnitude of  $2^t$ , where  $t$  is the security parameter, to make a brute force search (for  $n$  and  $k$ ) infeasible.

**Remark 3.2.2.** Alice has to take care that  $H$  and  $HM$  do not commute. Assume that  $H$  and  $HM$  commute, it is

$$\begin{aligned} Z &= h(m) \cdot H^{-n-k}(HM)^k H^n \\ &= h(m) \cdot H^{-k}(HM)^k \\ &= h(m) \cdot b. \end{aligned}$$

Hence, it adds up to calculate a new  $b'$  if an eavesdropper, Eve, wants a new  $Z'$  to impersonate herself as Alice. This is discussed in the Security 3.2.3 under (I) 1.

**Protocol 4**, the signature with  $G$  a semigroup of  $3 \times 3$  matrices over the group ring  $\mathbb{F}_7[A_5]$ , is summarized in Table 3.4 (page 83).

Table 3.4.: Summary of **Protocol 4**: Signature with a semigroup of  $3 \times 3$  matrices over  $\mathbb{F}_7[A_5]$

<b>Public Parameters</b>	
$G$ the semigroup of $3 \times 3$ matrices with entries in $\mathbb{F}_7[A_5]$ , an invertible $H \in G$ for the automorphism $\rho_H$ and a qualified hash function $h$ .	
Alice	Bob
Choose $n \in \mathbb{N}$ and $M \in G$ privately. Compute $(a, \rho_H^n) := (M, \rho_H)^n$ with $a := \rho_H^{n-1}(M) \cdot \rho_H^{n-2}(M) \cdots \rho_H(M) \cdot M$ $\quad = H^{-n}(HM)^n$ . Take care that $a^{-1} \notin G$ and that $H$ and $HM$ do not commute.	
Public Key: $a$	
Choose message $m$ and compute value $h(m) \in G$ . Pick an ephemeral key $k$ and compute $(b, \rho_H^k) := (M, \rho_H)^k$ with $b := \rho_H^{k-1}(M) \cdot \rho_H^{k-2}(M) \cdots \rho_H(M) \cdot M$ $\quad = H^{-k}(HM)^k$ . Compute $Z := h(m) \cdot \rho_H^n(b) = h(m) \cdot H^{-n-k}(HM)^k H^n$ .	
Signature: $(k, b, Z, m)$	
	Compute $(a, x) \cdot (b, \rho_H^k) = (\underbrace{\rho_H^k(a) \cdot b}_{=: E}, x \cdot \rho_H^k)$ , it is $E = H^{-(k+n)}(HM)^{n+k}$ . Prove $Z \cdot a = h(m) \cdot H^{-n-k}(HM)^{k+n}$ $\quad = h(m) \cdot E$ .

**Security 3.2.3.** The eavesdropper, Eve, knows Alice's public key  $a = H^{-n}(HM)^n$ . Eve wants to impersonate herself as Alice, that is, everyone should think that Eve's new message  $m'$  comes from Alice. Assume that Eve knows the signature  $S = (k, b, Z, m)$ .

(I) Eve chooses a new key  $k'$ :

She chooses new parameters  $(k', b', Z', m')$  where  $m'$  is the new message.

1. She has to calculate a new  $b'$ .

a) She needs to know the element  $M \in G$  which is one of Alice's secrets. She can get

$M$  from

$$\begin{aligned} H^{-1} \cdot \sqrt[k]{H^k \cdot b} &= H^{-1} \cdot \sqrt[k]{H^k \cdot H^{-k}(HM)^k} \\ &= H^{-1} \cdot \sqrt[k]{(HM)^k} \\ &= M. \end{aligned}$$

The difficulty here is to take the  $k$ th root from the element  $(HM)^k$ . This is a difficult problem in a finite semigroup of  $3 \times 3$  matrices over the group ring  $\mathbb{F}_7[A_5]$ .

If it was easy to calculate the correct  $k$ th root from  $(HM)^k$ , Eve could calculate the element  $b' = H^{-k'}(HM)^{k'}$ .

- b) Alternatively she uses a new  $k'$  with the property  $k' := k \cdot s$ , with  $s \in \mathbb{N}$ ,  $s > 1$ . Now, it is, with  $b = H^{-k}(HM)^k$ ,

$$\begin{aligned} u &:= (H^k \cdot b)^s = ((HM)^k)^s \\ &= (HM)^{k \cdot s} \\ &= (HM)^{k'} \end{aligned}$$

and it is  $b' = H^{-k'} \cdot u = H^{-k'} \cdot (HM)^{k'}$ . To prevent this, Alice and Bob could agree that Alice uses only **prime numbers for the ephemeral keys**  $k$ . If Bob gets a signature with  $k$  not a prime number he recognizes that Eve tried such an attack.

- c) Suppose, Eve knows several signatures

$$\begin{aligned} S_1 &= (k_1, b_{k_1}, Z_{k_1}, m_1), \\ S_2 &= (k_2, b_{k_2}, Z_{k_2}, m_2), \\ &\vdots \\ S_u &= (k_u, b_{k_u}, Z_{k_u}, m_u), \end{aligned}$$

with pairwise different ephemeral keys  $k_i$ . She can use the element  $b_{k_i} = H^{-k_i}(HM)^{k_i}$  to get

$$T_{k_i} := H^{k_i} \cdot b_{k_i} = (HM)^{k_i}.$$

It is

$$T_{k_i+k_j} = T_{k_i} \cdot T_{k_j} = (HM)^{k_i+k_j}.$$

The new  $b_{k_i+k_j}$  is now

$$b_{k_i+k_j} = H^{-(k_i+k_j)} \cdot T_{k_i+k_j} = H^{-(k_i+k_j)} \cdot (HM)^{k_i+k_j}.$$

In general Eve can calculate every  $b_{k'}$  with

$$k' = \sum_{i=1}^u \alpha_i \cdot k_i, \quad \text{with } \alpha_i \in \mathbb{N} \cup \{0\}.$$

If it is claimed that Alice's private key  $a$  has no inverse, then  $M$  cannot have an inverse; hence  $HM$  has no inverse. Therefore,  $\alpha_i$  cannot be a negative number. Thus, Eve can calculate  $b'_k$  whereby every new  $k'$  is always greater than the smallest number  $k_i$ .

A possible counter-measure is that Alice chooses at each new signature a lesser new ephemeral key than she uses for the previous signature. This leads to the problem, that Alice can just perform, with her private key  $n$ , a finite number of signatures, which depend on her first ephemeral key  $k_1$ .

2. After she has a new  $b'$  she needs a new element  $Z' = h(m') \cdot \rho_H^n(b')$ . There are two possibilities:

a) Eve tries to recover  $n$  from the public element  $a = H^{-n}(HM)^n$ . Note that Eve only knows the element  $HM$  if she can take the  $k$ -th root of the element  $(HM)^k$  (see above 1. a)).

As said in [HKKS13], a special case of this problem, where  $H = I$ , is the **discrete logarithm problem** for matrices over  $\mathbb{F}_7[A_5]$ . This problem is hard; it is addressed in [KKS13] in more detail.

M. Habeeb, D. Kahrobaei, C. Koupparis and V. Shpilrain also analyze whether or not any information about the private exponent  $n$  is leaked from transmission, that is, from the fact that Eve knows  $a = H^{-n}(HN)^n$ . That is the question: for a random exponent  $n$ , how different is the matrix in the first component of  $(M, \rho_H)^n = (H^{-n}(HM)^n, \rho_H^n)$  from  $N$ , where  $N$  is a random matrix? They find out, that no information about a private exponent  $n$  is revealed from the public element  $a = H^{-n}(HM)^n$  (see [HKKS13, Chapter 7]).

b) She does not know the secret  $n$ , therefore she has to calculate  $Z'$  in another way. Eve knows that Bob will verify the signature by the proof of the following equation

$$Z' \cdot a = h(m') \cdot \rho_H^{k'}(a) \cdot b'.$$

She can calculate  $Z'$  as

$$Z' = h(m') \cdot \rho_H^{k'}(a) \cdot b' \cdot a^{-1}$$

if the inverse of the element  $a$  exists.

Therefore, to prevent an attack (I) from Eve, Alice should assure that her **public element  $a$  has no inverse**. Hence, she can create the **signature only in a semigroup**. The element  $a = H^{-n}(HM)^n$  has no inverse if the matrix  $M$  is not invertible.

(II) Eve uses the same key  $k$ :

Eve chooses a new message  $m'$ . The elements  $k$  and  $b$  are the same. She only needs a new element  $Z'$ . Hence, she calculates

$$\begin{aligned} Z' &= h(m') \cdot (h(m))^{-1} \cdot Z \\ &= h(m') \cdot (h(m))^{-1} \cdot h(m) \rho_H^n(b) \\ &= h(m') \cdot \rho_H^n(b). \end{aligned}$$

Therefore, it is very easy for Eve to make everyone believe that her message  $m'$  comes from Alice. Alice and Bob could take care that **every ephemeral  $k$  is used only once**.

(III) Eve's information from  $Z$ :

We get two situations.

1. Let us first take a look at the situation if she wants to get the private key  $n$  with the help

from  $Z$ . Note that the hash function  $h$  is public. It is  $Z = h(m) \cdot \rho_H^n(b)$  and it follows

$$\begin{aligned} A &:= H^k \cdot (h(m))^{-1} \cdot Z = H^k \cdot \rho_H^n(b) \\ &= H^k \cdot H^{-n-k} (HK)^k H^n \\ &= H^{-n} \underbrace{(HM)^k}_{=:B} H^n. \end{aligned}$$

Eve knows  $B$  from  $H^k \cdot b = H^k \cdot H^{-k} (HM)^k = B$  and she can get  $x := H^n$  if she solves the **conjugations search problem**, that is: given two conjugate elements  $A, B \in G$ , find a particular element  $x \in G$ , such that  $x^{-1} B x = A$ . Suppose that she solves this problem and gets  $H^n$ , she then has to solve the **discrete logarithm problem** for matrices over  $\mathbb{F}_7[A_5]$ , namely recover  $n$  from  $H$  and  $H^n$ . This problem is hard (see [KKS13] for more details).

2. Suppose Eve knows several signatures

$$\begin{aligned} S_1 &= (k_1, b_{k_1}, Z_{k_1}, m_1), \\ S_2 &= (k_2, b_{k_2}, Z_{k_2}, m_2), \\ &\vdots \\ S_u &= (k_u, b_{k_u}, Z_{k_u}, m_u), \end{aligned}$$

with pairwise different ephemeral keys  $k_i$ .

With  $Z_{k_j} = h(m_j) \cdot H^{-n-k_j} (HM)^{k_j} H^n$  follows

$$X_{k_j} := H^{k_j} \cdot (h(m_j))^{-1} Z_{k_j} = H^{-n} (HM)^{k_j} H^n.$$

With very similar deliberations as in (I) 1. b) and c) we have:

a) Eve chooses a new  $k'$  with the property  $k' := k_i \cdot s$ , with  $s \in \mathbb{N}$ ,  $s > 1$ . It is

$$\begin{aligned} X_{k'=k_i \cdot s} &:= (X_{k_i})^s = \left( H^{-n} (HM)^{k_i} H^n \right)^s \\ &= H^{-n} (HM)^{k_i \cdot s} H^n \\ &= H^{-n} (HM)^{k'} H^n. \end{aligned}$$

For this  $k'$  she can get  $Z_{k'}$ , for the signature  $(k', b_{k'}, Z_{k'}, m')$ , with

$$\begin{aligned} Z_{k'} &= h(m') \cdot H^{-k'} \cdot X_{k'} \\ &= h(m') \cdot H^{-k'-n} (HM)^{k'} H^n. \end{aligned}$$

To prevent this, Alice and Bob could agree that Alice uses only **prime numbers for the ephemeral keys**  $k$ . If Bob gets a signature with  $k$  not a prime number he recognizes that Eve tried such an attack.

b) As above in a) it is

$$X_{k_j} := H^{k_j} \cdot (h(m_j))^{-1} Z_{k_j} = H^{-n} (HM)^{k_j} H^n.$$

It follows

$$X_{k_j+k_i} := X_{k_j} \cdot X_{k_i} = H^{-n} (HM)^{k_j+k_i} H^n.$$

The new  $Z_{k_j+k_i}$  for the signature  $(k_j + k_i, b_{k_j+k_i}, Z_{k_j+k_i}, m')$  is now

$$\begin{aligned} Z_{k_j+k_i} &= h(m') \cdot H^{-(k_j+k_i)} \cdot X_{k_j+k_i} \\ &= h(m') \cdot H^{-(k_j+k_i)-n} (HM)^{k_j+k_i} H^n. \end{aligned}$$

In general Eve can calculate every  $Z_{k'}$  with

$$k' = \sum_{i=1}^u \alpha_i \cdot k_i, \quad \text{with } \alpha_i \in \mathbb{N} \cup \{0\}.$$

If it is claimed that the private key  $a$  from Alice has no inverse, then  $M$  cannot have an inverse; hence  $HM$  has no inverse. Therefore,  $\alpha_i$  cannot be a negative number. Thus, Eve can calculate  $Z_{k'}$  whereby every new  $k'$  is always greater than the smallest number  $k_i$ .

A possible counter-measure is that Alice chooses at each new signature a lesser new ephemeral key than she uses for the previous signature. This leads to the problem, that with her private key  $n$ , Alice can only perform a finite number of signatures depending on her first ephemeral key  $k_1$ .

If Eve tries to impersonate herself as Alice with the information from (III) she also needs the corresponding  $b_{k'}$ , which is discussed in (I) 1.

The counter-measure from Alice against Eve's attacks (II) and (III) should be, to determine, that the **image of the hash function  $h$  is only the non-invertible matrices** in the semigroup  $G$ . Hence, Eve does not know the element  $\rho_H^n(b_{k_j})$  and therefore she cannot use  $X_{k_j}$  for an attack.

### 3.3. Security and ongoing research about the HKKS-key exchange protocol

There is an ongoing research about the HKKS-key exchange protocol, which also affects the ElGamal like key exchange of Section 3.1. In this section we give an overview of this research.

Paper [KMU14]:

As mentioned above M. Kreuzer, A. D. Maysnikov and A. Ushakov published the paper [KMU14] in which they describe a linear algebra attack on the HKKS-scheme when the proposed semigroup  $G = \text{Mat}(3, \mathbb{F}_7[A_5])$  is used as it is suggested in [HKKS13, Chapter 6]. This is the semigroup of  $3 \times 3$  matrices over the group ring  $\mathbb{F}_7[A_5]$ , where  $A_5$  is the alternating group on 5 elements and  $\mathbb{F}_7$  is the field with 7 elements (here, more precisely, it is  $\mathbb{F}_7^* = (\mathbb{Z}/7\mathbb{Z})^*$ ). In addition they use an extension of the semigroup  $G$  by an inner automorphism  $\varphi_H$ , which is conjugation by a matrix  $H \in \text{GL}(3, \mathbb{F}_7[A_5])$ .

In their attack, they use an embedding of  $\text{Mat}(3, \mathbb{F}_7[A_5])$  into  $\text{Mat}(180, \mathbb{F}_7)$  and they are able to reconstruct the key, which is  $H^{-n-k}(HM)^{n+k}$ , without the knowledge of the private exponents  $n$  and  $k$  of Alice and Bob, respectively. For this it is sufficient to find two matrices  $l, r \in G$  satisfying the following system of matrix equations:

$$l \cdot H = H \cdot l \tag{3.1}$$

$$r \cdot (HM) = (HM) \cdot r \tag{3.2}$$

$$a = lr, \tag{3.3}$$

with  $H$  and  $M$  public parameters and  $a$  the public element from Alice. If matrices  $l$  and  $r$  are known, which satisfy the equations (3.1)-(3.3), then the shared key is computable as

$$\begin{aligned} l \cdot b \cdot r &= l \cdot H^{-k}(HM)^k \cdot r \\ &= H^{-k} \cdot l \cdot r \cdot (HM)^k \\ &= H^{-k} \cdot H^{-n}(HM)^n \cdot (HM)^k = H^{-k-n}(HM)^{k+n}, \end{aligned}$$

with  $b = H^{-k}(HM)^k$ , the sent element from Bob. M. Kreuzer, A. D. Myasnikov and A. Ushakov observed that the system (3.1)-(3.3) has at least one solution with  $l \in \text{GL}(3, \mathbb{F}_7[A_5])$ , that means, with an invertible matrix  $l = H^{-n}$  and  $r = (HM)^n$ . Hence, instead of solving the matrix equations (3.1)-(3.3) it is sufficient to solve the system

$$\begin{aligned} \ell \cdot H &= H \cdot \ell \\ r \cdot (HM) &= (HM) \cdot r \\ \ell a &= r, \end{aligned}$$

and recover the matrix  $l$  from the equation  $\ell \cdot l = 1$ .

They get a system of linear equations over  $\mathbb{F}_7$  for  $\ell$  and explain how they are able to find  $\ell$ , for more details see [KMU14].

Such an attack also violates the ElGamal like cryptosystem as explained in Table 3.3 (page 80). If  $H^{-k-n}(HM)^{k+n}$  is known, then the message  $m$  in the ElGamal like cryptosystem is reconstructible, because

$$(H^{-k-n}(HM)^{k+n})^{-1} \cdot c_2 = m.$$

Thus, this attack should also be possible for Example 3.1.4, which is in more details explained in Table 3.3 (page 80). The elements which Eve needs to know for such an attack, and hence to generate  $H^{-k-n}(HM)^{k+n}$ , are

$$\begin{aligned} H^{-n}(HM)^n, \\ H^{-k}(HM)^k, \\ H \text{ and } HM, \end{aligned}$$

which are also public in the ElGamal like cryptosystem.

It is important to see, that there is another algorithm that recovers the private key without solving the principal algorithmic problems on which the security assumptions are based.

Recall, that the security assumptions are the following.

1. It is computationally hard to retrieve the key  $K = H^{-n-k}(HM)^{n+k}$  from the quadruple  $(H, M, H^{-k}(HM)^k, H^{-n}(HM)^n)$ . Thus, they have to take care, that  $H$  and  $HM$  do not commute, because otherwise  $K$  is just a product of  $H^{-k}(HM)^k$  and  $H^{-n}(HM)^n$ .
2. It is hard to recover  $k$  from  $H^{-k}(HM)^k$  or  $n$  from  $H^{-n}(HM)^n$ , respectively. A special case of this problem, where  $H = I$ , is the discrete logarithm problem for matrices over  $\mathbb{F}_7[A_5]$ , this problem is hard, see [KKS13].
3. For a random exponent  $n$ , how different is the matrix of the first component of  $(M, \varphi_H)^n$  from  $N$ , where  $N$  is a random matrix in  $\text{Mat}(3, \mathbb{F}_7(A_5))$ ? In [HKKS13, Chapter 7] it is explained, that they are indistinguishable.
4. How different is the final secret key from a random matrix? This is in more details the question: if  $n$  and  $k$  are the secret random keys from Alice and Bob how different is the matrix of the first component of  $(M, \varphi_H)^{k+n}$  from the matrix of the first component of



$(M, \varphi_H)^q$ , where  $q$  is of the same bit size as  $n + k$ . In [HKKS13, Chapter 7] it is explained, that they are indistinguishable.

Paper [KLS15]:

D. Kahrobaei, H. T. Lam and V. Shpilrain suggest in the paper [KLS15] for the platform for the HKKS-scheme the semigroup of matrices over a Galois field of characteristic 2, more specifically over  $\mathbb{F}_{2^{127}}$ , thus they reduce the key size and speed up the computation quite a bit. Furthermore, the automorphism that they use in their paper [KLS15] is not inner, but a composition of an inner automorphism with the endomorphism that raises each entry of a given matrix to the power of 4.

If one uses just inner automorphisms and matrices over a field they suggest an extra “tweak” of the cryptographic protocol to avoid a linear algebra attack as explained in [KMU14].

We explain in short their suggested variations.

1. Let  $G$  be the semigroup of  $2 \times 2$  matrices over  $\mathbb{F}_{2^{127}}$ . They suggest for  $\mathbb{F}_{2^{127}}$  the factor algebra  $\mathbb{Z}_2[x]/\langle p(x) \rangle$ , with  $\mathbb{Z}_2 := \mathbb{Z}/2\mathbb{Z}$  and  $\langle p(x) \rangle$  is the ideal of the polynomial algebra  $\mathbb{Z}_2[x]$  generated by the irreducible polynomial  $p(x) = x^{127} + x^{63} + 1$ . Elements of  $\mathbb{F}_{2^{127}}$  are therefore here polynomials of degree at most 126 over  $\mathbb{Z}_2$ . They use an extension of the semigroup  $G$  by endomorphism  $\varphi$ , which is a composition of a conjugation by a matrix  $H \in \text{GL}(2, \mathbb{F}_{2^{127}})$  with the endomorphism  $\psi$  that rises each entry of a given matrix to the power of 4. The composition is such that  $\psi$  is applied first, followed by conjugation. Thus, for any matrix  $M \in G$  and for any integer  $k \geq 1$ , we have

$$\begin{aligned} \varphi(M) &= H^{-1}\psi(M)H \quad \text{and} \\ \varphi^k(M) &= H^{-1}\psi(H^{-1})\psi^2(H^{-1}) \dots \psi^{k-1}(H^{-1})\psi^k(M)\psi^{k-1}(H) \dots \psi(H)\psi^2(H)H. \end{aligned}$$

The rest of the cryptographic protocol stays the same as the HKKS-protocol.

2. If the public automorphism  $\varphi$  is just conjugated by a public matrices  $H$ , the transmitted matrices are then  $H^{-n}(HM)^n$  and  $H^{-k}(HM)^k$ . Thus, this cryptographic protocol is vulnerable to linear algebra attacks, if the matrices involved in this cryptographic protocol are over a field or over a ring that can itself be embedded in a ring of matrices over a field. The attacker Eve is looking for matrices  $X$  and  $Y$ , such that

$$\begin{aligned} XH &= HX \\ Y(HM) &= (HM)Y \\ XY &= H^{-n}(HM)^n. \end{aligned}$$

The first two matrix equation translate into a system of linear equations in the entries of  $X$  and  $Y$  over the ground field, whereas the last one does not. However, if  $X$  is invertible, then the last matrix equation can be rewritten as  $Y = X^{-1}H^{-n}(HM)^n$ , and this translates into a system of linear equations in the entries of  $X^{-1}$  and  $Y$ . Thus, upon replacing the first matrix equation  $XH = HX$  by the equivalent  $X^{-1}H = HX^{-1}$ , Eve ends up with a system of linear equations in the entries of  $X^{-1}$  and  $Y$  over the ground field. After solving this system and finding  $X$  and  $Y$ , Eve can recover the shared secret key  $K$  from the public transmissions as follows:

$$\begin{aligned} X(H^{-k}(HM)^k)Y &= H^{-k}XY(HM)^k = H^{-k}H^{-n}(HM)^n(HM)^k \\ &= H^{-n-k}(HM)^{n+k} = K \end{aligned}$$

This kind of attack may also work if the platform is a semigroup consists of matrices not over a field, but over a ring that can itself be embedded in a ring of matrices over a field,

see for example [KMU14].

Thus, they give a “tweak” to avoid such kinds of attacks in semigroups. In addition to the secret key  $n$  Alice also selects a private matrix  $R \neq 0 \in G$ , such that  $R(HM) = 0$ , whereby  $0$  denotes the zero matrix. Such a matrix  $R$  exists, because  $HM$  is not invertible. Recall, that  $G$  is a semigroup. Bob does the same, he selects in addition to his private ephemeral key  $k$  a matrix  $S \neq 0 \in G$  with  $S(HM) = 0$ . Now, Alice sends

$$A = H^{-n}(HM)^n + R$$

to Bob and he sends

$$B = H^{-k}(HM)^k + S$$

to Alice. The step to recover the common secret key stays the same for Alice and Bob.

Alice calculates  $(B, x) \cdot (H^{-n}(HM)^n, \varphi^n)$  where the first component is

$$H^{-n-k}(HM)^{n+k} + (H^{-n}SH^n) \cdot (H^{-n}(HM)^n) = H^{-n-k}(HM)^{n+k} = K,$$

because  $S(HM) = 0$ . Bob computes  $(A, y) \cdot (H^{-k}(HM)^k, \varphi^k)$  where the first component is

$$H^{-n-k}(HM)^{n+k} + (H^{-k}RH^k) \cdot (H^{-k}(HM)^k) = H^{-n-k}(HM)^{n+k} = K$$

because  $R(HM) = 0$ .

These variations are not interesting for the ElGamal like cryptosystem, because they use semigroups and the cryptosystem needs a group. Moreover, the next paper [Rom15] by V. Roman’kov gives a linear decomposition attack which breaks also these two variations.

Paper [Rom15]:

V. Roman’kov shows in [Rom15] that a linear decomposition attack based on the decomposition method introduced by the author, V. Roman’kov, in monography [Rom13a] and paper [Rom13b] works by finding exchange keys in the both main cryptographic protocols in [HKKS13] and [KLS15]. The attack works when the platform groups are linear.

He shows, that in this case, contrary to the common opinion (and some explicitly stated security assumptions), one does not need to solve the underlying algorithmic problems (discrete logarithm problem and Diffie-Hellman problem) to break the scheme, that means, there is another algorithm that recovers the keys without solving the principal algorithmic problem on which the security assumptions were first based. This changes completely the understanding of security of this cryptographic protocol. The efficacy of the attack depends on the platform group, so it requires a specific analysis in each particular case. In general V. Roam’kov mentioned that one can only state that the attack is in polynomial time in the size of the data, when the platform and related groups are given together with their linear representations. In many other cases we can effectively use known linear representations of the groups under considerations.

Main points in Roman’kovs paper:

1. He shows for the HKKS-key exchange protocol, that the shared secret key  $K$  can be computed in the case when  $G$  is a multiplicative subgroup of a finite dimensional algebra  $\mathbf{A}$  over a field  $\mathbb{F}$  and the endomorphism  $\phi$  is extended to an endomorphism of the underlying vector space  $V$  of  $\mathbf{A}$ . Furthermore, we assume that the basic field operations in  $\mathbb{F}$  are efficient in particular they can be performed in polynomial time in the size of the elements, for example,  $\mathbb{F}$  is finite. In all the particular cryptographic protocols considered in the paper [Rom15] the field  $\mathbb{F}$  satisfies all these conditions.

Using Gauß elimination Roman'kov showed that one can effectively find a maximal linearly independent subset  $L$  of the set  $\{a_0, a_1, \dots, a_\ell, a_{\ell+1}, \dots\}$ , where it is  $a_0 = g$  and  $a_\ell = \phi^{\ell-1}(g) \cdots \phi(g) \cdot g$  for  $\ell \geq 1$ . Indeed, suppose that  $\{a_0, a_1, \dots, a_\ell\}$  is a linear independent set but  $a_{\ell+1}$  can be presented as a linear combination of the form

$$a_{\ell+1} = \sum_{i=0}^{\ell} \lambda_i a_i \quad \text{for } \lambda_i \in \mathbb{F}.$$

Suppose by induction that  $a_{\ell+j}$  can be presented as above for every  $j \leq t-1$ . In particular

$$a_{\ell+t-1} = \sum_{i=0}^{\ell} \mu_i a_i \quad \text{for } \mu_i \in \mathbb{F}.$$

Then

$$\begin{aligned} a_{\ell+t} &= \phi(a_{\ell+t-1}) \cdot g = \sum_{i=0}^{\ell} \mu_i \phi(a_i) \cdot g \\ &= \sum_{i=0}^{\ell} \mu_i a_{i+1} = \mu_\ell \lambda_0 a_0 + \sum_{i=0}^{\ell-1} (\mu_i + \mu_\ell \lambda_{i+1}) a_{i+1}. \end{aligned}$$

Thus,  $L = \{a_0, a_1, \dots, a_\ell\}$ . In particular, we can effectively compute

$$a_k = \sum_{i=0}^{\ell} \eta_i a_i \quad \text{for } \eta_i \in \mathbb{F}. \quad (3.4)$$

Then

$$a_{k+n} = \phi^n(a_k) \cdot a_n \quad (3.5)$$

$$= \sum_{i=0}^{\ell} \eta_i \phi^n(a_i) \cdot a_n = \sum_{i=0}^{\ell} \eta_i \phi^i(a_n) \cdot a_i. \quad (3.6)$$

Note that all data on the right hand side is known. Thus, we get the shared key  $K = a_{n+k}$ . Hence, in the case, where  $G$  is in  $Mat(3, \mathbb{F}_7[A_5])$ , there is a polynomial time algorithm to find the shared key  $K$  from the public data.

2. Using matrices over a Galois Field and extensions by special endomorphisms [KLS15]:  
V. Roman'kov showed that this cryptographic protocol, even with the endomorphism  $\psi$  that rises each entry of a given matrix to the power of 4 and the composition is just that  $\psi$  is applied first, followed by conjugation, can be attacked by the linear decomposition attack as above.
3. In [KLS15] D. Kahrobaei, H. T. Lam and V. Shpilrain showed that the version of [HKKS13] is vulnerable to linear algebra attacks (which was also mentioned in [KMU14]). Roman'kov showed in his paper, that his linear decomposition attack is very simple in contrast to the linear algebra attack described in [KMU14].
4. In [KLS15] they gave a “tweak” to avoid the linear algebra attack. Unfortunately this cryptographic protocol is also vulnerable against the linear decomposition attack. This is also described in [Rom15].

5. Note, in the attacks by Roman'kov the secret keys  $k$  or  $n$  are not needed to recover the secret key  $K$ .

Paper [KS16]:

D. Kahrobaei and V. Shpilrain shift their focus in [KS16] to select an optimal platform (semi)group, in terms of security and efficiency.

They suggest the group  $G = F_r/F_r^{p^2} \cdot \gamma_{c+1}(F_r)$ . This group, being a nilpotent  $p$ -group, is finite. We recall the definitions of nilpotent groups and  $p$ -groups.

Let  $F_r$  be the free group generated by the free generating set  $\{x_1, x_2, \dots, x_r\}$ , that is,

$$F_r = \langle x_1, x_2, \dots, x_r \mid \ \rangle.$$

The normal subgroup  $F_r^p$  is generated (as a group) by all elements of the form  $g^p$ ,  $g \in F_r$ . In the factor group  $F_r/F_r^p$  every nontrivial element has order  $p$  (if  $p$  is a prime number). More generally, if  $n \geq 2$  is an arbitrary integer, then the order of any element of  $F_r/F_r^n$  divides  $n$ .

The other normal subgroup, which is needed, is somewhat less straightforward to define.

Let  $[a, b]$  denote the commutator with  $[a, b] = a^{-1}b^{-1}ab$ . Then, inductively, let  $[y_1, y_2, \dots, y_{c+1}]$  denote  $[[y_1, y_2, \dots, y_c], y_{c+1}]$ . For a group  $G$ , denote by  $\gamma_c(G)$  the (normal) subgroup of  $G$  generated (as a group) by all elements of the form  $[y_1, y_2, \dots, y_c]$ . If  $\gamma_{c+1}(G) = \{1\}$ , we say that the group  $G$  is nilpotent of nilpotency class  $c$ .

The factor group  $F_r/\gamma_{c+1}(F_r)$  is called **the free nilpotent group** of nilpotency class  $c$ . This group is infinite. The group  $G = F_r/F_r^{p^2} \cdot \gamma_{c+1}(F_r)$ , which they recommend as platform group, is finite, the order depends on  $p, c$  and  $r$ . For more details see [KS16].

The number  $p$  should be large enough to make the dimension of linear representations of  $G$  so large that a linear algebra attack would be infeasible. As they mentioned in the introduction, a faithful representation of a finite  $p$ -group, with at least one element of order  $p^n$ , as a group of matrices over a finite field of characteristic  $p$  is of dimension at least  $1 + p^{n-1}$  ([Jan70]), so in their case it is of dimension at least  $1 + p$ . Thus, if  $p$  is, say, a 100-bit number, a linear algebra attack is already infeasible. For efficiency reasons it seems better to keep  $c$  and  $r$  fairly small (in particular they suggest  $c = 2$  or  $3$ ).

# Chapter 4

## Combinatorial group theory

This chapter yields the mathematical background for **Protocol 6** to **Protocol 12**, which are based on combinatorial group theory. Combinatorial group theory is the branch of algebra which studies groups via their presentations, that is, via their description with generators and relators. All seven cryptographic protocols (**Protocol 6** to **Protocol 12**) use finitely generated free groups as platform groups; therefore, we start with a detailed introduction of free groups and group presentations.

Among free groups we introduce Nielsen transformations, Nielsen reduced sets and corresponding theory, which will be used later on for the cryptographic protocols. Nielsen transformations are a linear technique to study free groups and general infinite groups.

Afterwards, we explain several fundamental problems in group theory, which could be used for cryptology. One problem is the extended word problem, also called membership problem, another related problem is the constructive membership problem, both problems play a role for the security of the newly developed cryptographic protocols.

In addition **Protocol 6** to **Protocol 12** use automorphisms on finitely generated free groups. These automorphisms can be generated with the help of Nielsen transformations or alternatively with so called Whitehead-Automorphisms. Therefore, we close this chapter introducing Whitehead-Automorphisms. With the help of these automorphisms we could develop an approach for choosing automorphisms randomly of the automorphism group  $Aut(F)$ , with  $F$  a finitely generated free group  $F$ .

The books [CgRR08], [LS77] and [MKS66] are the basis for this chapter. The reader should be familiar with the basics of groups as it is presented in a course about algebra (see for instance [JS06]).

### 4.1. Free groups and group presentations

Let  $G$  be a group, let  $X \subseteq G$  be a subset and  $X^{-1} := \{x^{-1} \mid x \in X\}$ , the set of the inverse elements of the elements in  $X$ . The subgroup generated by  $X$  is labeled with  $\langle X \rangle$  and consists of all finite products of elements from  $X^{\pm 1} := X \cup X^{-1}$ . It is

$$\langle X \rangle = \{x_1 x_2 \cdots x_m \mid x_1, x_2, \dots, x_m \in X^{\pm 1}, m \in \mathbb{N}\}.$$

We call a term of the form  $w = x_1 x_2 \cdots x_m$ , with  $x_i \in X^{\pm 1}$  a word in  $X$  or just a **word**. Each word corresponds to an element in  $G$ . The identity in  $G$  corresponds to the empty word and is labeled with 1.

**Definition 4.1.1.** (universal property) [CgRR08]

Let  $X$  be a nonempty set,  $F$  a group and  $\iota : X \rightarrow F$  an injective map. The group  $F$  - more precisely the tuple  $(F, \iota)$  - is named **free on  $X$** , if there exists for every group  $G$  and every map

$f : X \rightarrow G$  an uniquely defined homomorphism  $\varphi : F \rightarrow G$ , such that  $f = \varphi \circ \iota$ , that means the following diagram (Figure 4.1)

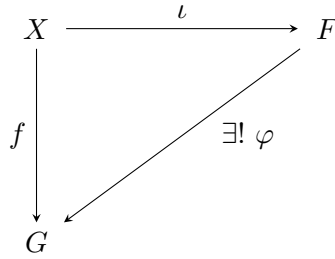


Figure 4.1.: Commuting diagram

commutes.

At most we consider  $X$  as subset of the group  $F$  with  $\iota$  as inclusion. Then, the set  $X$  is named **free generating set** of  $F$ .

**Example 4.1.2.** The set  $\{1\}$  is a free generating set for the group  $(\mathbb{Z}, +)$ .

**Theorem 4.1.3.** [CgRR08, Satz 1.2]

Let  $F_1$  and  $F_2$  - more precisely  $(F_1, \iota_1)$  and  $(F_2, \iota_2)$  - be free on  $X$ . Then there exists an isomorphism  $\varphi : F_1 \rightarrow F_2$  with  $\varphi \circ \iota_1 = \iota_2$ , that means,  $F_1$  is unique up to isomorphisms.

**The free group  $F(X)$ :**

Next, we show, that for each set  $X$  exists exactly one group  $F$ , up to isomorphisms (Theorem 4.1.3), which is free on  $X$ . This group  $F$  is constructed as follows:

If  $X = \emptyset$ , then  $F(X) = \{1\}$ .

Let  $X \neq \emptyset$  with  $X^{-1}$  as above, it is  $X \cap X^{-1} = \emptyset$  and let  $x \mapsto x^{-1}$  be a bijection from  $X$  to  $X^{-1}$ . Recall, that  $X^{\pm 1} := X \cup X^{-1}$ . We denote the set of all finite sequences  $(x_1, x_2, \dots, x_m)$ ,  $x_i \in X^{\pm 1}$  and  $m \geq 0$ , with  $M(X)$ . If  $m = 0$ , we get the empty sequence. We define an associative multiplication on  $M(X)$  by concatenation, that is,

$$(x_1, x_2, \dots, x_m) \cdot (y_1, y_2, \dots, y_{m'}) := (x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_{m'}).$$

The identity is the empty sequence, and is labeled with 1. The map  $X^{\pm 1} \rightarrow M(X)$  with  $x \mapsto (x)$  is injective and  $(x)$  is identified with  $x$ . Each element  $w \in M(X)$  is uniquely presented as a **word** (in  $X$ ), that is, as a product of elements from  $X^{\pm 1}$ :

$$w = x_1^{\epsilon_1} x_2^{\epsilon_2} \cdots x_m^{\epsilon_m}, \quad \text{with } x_i \in X \text{ and } \epsilon_i = \pm 1,$$

whereby  $x_i^{+1}$  is identified with  $x_i$ .

The set  $M(X)$  is a monoid. It is called **free monoid** on  $X^{\pm 1}$ . The elements of  $X^{\pm 1}$  are called **letters**. A word  $w$  is called **reduced**, if  $x_i \neq x_{i+1}$  or  $x_i = x_{i+1}$  but  $\epsilon_i + \epsilon_{i+1} \neq 0$ , for  $1 \leq i < m$ ; the empty word is reduced. If  $w = x_1^{\epsilon_1} x_2^{\epsilon_2} \cdots x_m^{\epsilon_m}$  is not reduced, we say,  $w'$  ensued of  $w$  by **elementary reduction**, if  $w' = x_1^{\epsilon_1} x_2^{\epsilon_2} \cdots x_{i-1}^{\epsilon_{i-1}} x_{i+2}^{\epsilon_{i+2}} \cdots x_m^{\epsilon_m}$ , with  $x_i = x_{i+1}$  and  $\epsilon_i + \epsilon_{i+1} = 0$ .

We can write a reduced word  $w$  as

$$w = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_m^{\alpha_m}, \quad \text{with } x_i \in X, x_i \neq x_{i+1} \text{ and } \alpha_i \in \mathbb{Z} \setminus \{0\}.$$

We get an equivalence relation on  $M(X)$  while we put  $w \equiv w'$ , if  $w = w'$  or there is a finite sequence  $w = w_1, w_2, \dots, w_k = w'$ , such that either from  $w_{j+1}$  follows  $w_j$  or from  $w_j$  follows

$w_{j+1}$  by elementary reduction with  $1 \leq j < k$ . For every  $w \in M(X)$  there is always a reduced word  $w' \in M(X)$  with  $w \equiv w'$ . If  $w \equiv w'$  then  $uwv \equiv uw'v$  for all  $u, v \in M(X)$  and if  $w \equiv w'$  and  $v \equiv v'$ , then also  $vw \equiv v'w'$ . Hence, the multiplication on  $M(X)$  implies a multiplication on  $F(X) := M(X) / \equiv$ , the set of the equivalence classes of  $M(X)$  concerning  $\equiv$ , with

$$[u][v] := [uv],$$

whereby  $[w]$  denotes the class of the word  $w$ . The multiplication is associative (transmission of the quotient structure) and the identity is the class of the empty word.

The set  $F(X)$  is a group and  $F(X)$  is free on  $X$  (see [CgRR08, p. 5]).

**Corollary 4.1.4.** [CgRR08, Korollar 1.4]

Let  $G$  be a group with  $G = \langle X \rangle$ . Then  $G$  is isomorphic to a quotient group of  $F(X)$ .

**Definition 4.1.5.**

In general a group  $G$  is called a free group if there exists a set  $X$  with  $G \cong F(X)$ .

**Theorem 4.1.6.** (Normalformensatz)[CgRR08, Satz 1.5]

There is exactly one reduced word in each equivalence class. That means, for every word  $w$  exists exactly one reduced word  $w' = x_1^{\epsilon_1} x_2^{\epsilon_2} \cdots x_m^{\epsilon_m}$ ,  $\epsilon_i \in \{1, -1\}$ , with  $w \equiv w'$ . The uniquely defined number  $m$  is called the **free length** of  $w$ ; denoted with  $|w|$ .

**Group presentation via generators and relators:**

Let  $G$  be a group,  $X$  a set and  $\varphi : F(X) \twoheadrightarrow G$  be a group epimorphism. We call  $X$  a **generating set** for  $G$  under  $\varphi$  and  $\{\varphi(x) \mid x \in X\}$  a set of **generators** of  $G$ . It is

$$G = \langle \varphi(x) \mid x \in X \rangle.$$

We also call  $X$  just a generating set for  $G$ . The kernel  $\ker \varphi$  is called the set of **relators** of  $G$  under  $\varphi$ .

If  $u = x_1^{\epsilon_1} \cdots x_n^{\epsilon_n}$  and  $v = y_1^{\eta_1} \cdots y_m^{\eta_m}$ , with  $x_i, y_j \in X$ , such that  $uv^{-1} \in \ker \varphi$  and  $\varphi(x_i) = a_i$ ,  $\varphi(y_j) = b_j$ , then we call  $a_1^{\epsilon_1} \cdots a_n^{\epsilon_n} = b_1^{\eta_1} \cdots b_m^{\eta_m}$  a **relation** in  $G$ . Especially, if  $u \in \ker \varphi$ , then it is  $a_1^{\epsilon_1} \cdots a_n^{\epsilon_n} = 1$  a relation in  $G$ .

Now, let  $H$  be a group. Let  $S \subseteq H$ , we call the normal closure

$$\langle\langle S \rangle\rangle := \langle hsh^{-1} \mid s \in S, h \in H \rangle$$

of  $S$  in  $H$  the set of **consequences** of  $S$  in  $H$ .

A subset  $R \subseteq F(X)$  is called a set of **defining relators** of  $G$  (under  $\varphi$ ), if  $\ker \varphi$  is the set of consequences of  $R$ . The image of  $R$  under  $\varphi$  is called the corresponding set of **defining relations**.

We also call a relation  $\varphi(u) = \varphi(v)$  a **consequence** of a set of defining relators (or defining relations) if the corresponding relator  $uv^{-1}$  is a consequence of the defining relators.

Presentation:

A **presentation**  $\langle X \mid R \rangle^\varphi$  of  $G$  consist of a set  $X$ , a group epimorphism  $\varphi : F(X) \twoheadrightarrow G$  and a set of defining relators of  $G$  (under  $\varphi$ ). The group  $G$  is called **finitely presented**, if both sets  $X$  and  $R$  are finite.

Convention:

We write  $G = \langle X \mid R \rangle^\varphi$ , if  $\langle X \mid R \rangle^\varphi$  is a presentation of  $G$ . Usually we omit  $\varphi$ , especially if  $\varphi$  is the canonical mapping

$$\varphi : F(X) \twoheadrightarrow F(X) / \langle\langle R \rangle\rangle$$

or, if  $\varphi$  is injective on  $X$  (then it is  $X \subseteq G$ ).

Often, we replace the relator  $uv^{-1}$  by the relation  $u = v$ . We mix the words “relators” and “relations”, without getting misunderstandings. We mostly write

$$G = \langle X \mid r = 1, r \in R \rangle$$

for a presentation of  $G$ .

**Example 4.1.7.** 1. A free group  $F(X)$  has only relations which are consequences of  $x_i^{-1}x_i$  and  $x_ix_i^{-1}$ , hence  $R = \emptyset$  and we write  $F(X) = \langle X \mid \ \rangle$ .

2. A cyclic group  $\mathbb{Z}/p\mathbb{Z} = \langle x \mid x^p = 1 \rangle$ , with  $p$  a prime number.

3. The trivial group  $\{1\} = \langle x \mid x = 1 \rangle = \langle x, y \mid x = 1, xy^{-1} = 1 \rangle$ .

**Corollary 4.1.8.** [CgRR08, Korollar 3.4]

*Two groups with the same presentation are isomorph.*

Instead of  $F(X)$  we also write just  $F$  for a free group on a free generating set  $X$ . We denote a subgroup of  $F$  by  $F_U$  if it is generated by a free generating set  $U$ , whereby the elements in  $U$  are words in  $X$ .

**Remark 4.1.9.** Let  $X$  be the free generating set for the free group  $F = \langle X \mid \ \rangle$ . We call  $X$  also a **basis** of  $F$ . The cardinality  $|X|$  of a basis for a free group  $F$  is called the **rank** of the group. The rank for a free group is unique, because two free groups are isomorphic if and only if their basis have the same cardinality (see Theorem 4.3.7).

## 4.2. Nielsen transformations, Nielsen reduced sets and additional theory

Let  $F$  be a free group with free generating set  $X := \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ . We determine that words in  $X$  are reduced. Let  $w \in F$ , we write  $w \equiv w_1w_2 \cdots w_k$  if there are no cancellations between the words  $w_i$  and  $w_{i+1}$ ,  $1 \leq i < k$ , that means  $|w| = |w_1| + |w_2| + \cdots + |w_k|$ .

Let  $U := \{u_1, u_2, \dots, u_t\} \subset F$ ,  $t \geq 2$ , with  $u_i$  words in  $X$ .

**Definition 4.2.1.** An **elementary Nielsen transformation** on  $U = \{u_1, u_2, \dots, u_t\} \subset F$  is one of the following transformations

(T1) replace some  $u_i$  by  $u_i^{-1}$ ;

(T2) replace some  $u_i$  by  $u_iu_j$  where  $j \neq i$ ;

(T3) delete some  $u_i$  where  $u_i = 1$ .

In all three cases the  $u_k$  for  $k \neq i$  are not changed. A (finite) product of elementary Nielsen transformations is called a **Nielsen transformation**. A Nielsen transformation is called **regular** if it is a finite product of the transformations (T1) and (T2), otherwise it is called **singular**.

The regular Nielsen transformations generate a group (see for instance[CgRR08]). The set  $U$  is called **Nielsen equivalent** to the set  $V$ , if there is a regular Nielsen transformation from  $U$  to  $V$ . Nielsen equivalent sets  $U$  and  $V$  generate the same group, that is,  $\langle U \rangle = \langle V \rangle$ .



Each elementary Nielsen transformation  $(T1)$  and  $(T2)$  has an inverse which is a regular Nielsen transformation. It follows then, that the regular Nielsen transformations form a group which contains every permutation of the set  $U = \{u_1, u_2, \dots, u_t\}$  (see for instance [FGMRS14]).

Now, we agree on some notations. We write  $(T1)_i$  if we replace  $u_i$  by  $u_i^{-1}$  and we write  $(T2)_{i,j}$  if we replace  $u_i$  by  $u_i u_j$ ; if we want to apply the same Nielsen transformation  $(T2)$  consecutively  $\ell$ -times we write  $[(T2)_{i,j}]^\ell$  and hence replace  $u_i$  by  $u_i u_j^\ell$ . Thus, it is

$$\begin{aligned} (T1)_i & : (u_1, \dots, u_i, \dots, u_t) \rightarrow (u_1, \dots, u_i^{-1}, \dots, u_t), \\ (T2)_{i,j} & : (u_1, \dots, u_i, \dots, u_j, \dots, u_t) \rightarrow (u_1, \dots, u_i u_j, \dots, u_j, \dots, u_t), \\ [(T2)_{i,j}]^\ell & : (u_1, \dots, u_i, \dots, u_j, \dots, u_t) \rightarrow (u_1, \dots, u_i u_j^\ell, \dots, u_j, \dots, u_t). \end{aligned}$$

**Definition 4.2.2.**

A finite set  $U$  in  $F$  is called **Nielsen reduced**, if for any three elements  $v_1, v_2, v_3$  from  $U^{\pm 1}$  the following conditions hold:

(N0)  $v_1 \neq 1$ ;

(N1)  $v_1 v_2 \neq 1$  implies  $|v_1 v_2| \geq |v_1|, |v_2|$ ;

(N2)  $v_1 v_2 \neq 1$  and  $v_2 v_3 \neq 1$  implies  $|v_1 v_2 v_3| > |v_1| - |v_2| + |v_3|$ .

Recall,  $|v|$  denotes the **free length** of  $v \in F$ , that is, the number of letters from  $X^{\pm 1}$  in the freely reduced word  $v$ . We could also write  $|v|_X$ , if it is not clear from which set the letters of  $v$  are and we count the letters in  $v$  which are given as elements in  $X^{\pm 1}$ .

**Remark 4.2.3.** We say that any word  $w$ , not necessary reduced, with finitely many letters from  $X^{\pm 1}$  has **length**  $L$  if the number of letters occurring is  $L$ . The length of a word  $w$  is greater than or equal to the free length of the word  $w$ . For freely reduced words the length and the free length are equal. If a word  $w$  is not freely reduced then the length is greater than the free length of  $w$ .

**Proposition 4.2.4.** [CgRR08, Korollar 2.10]

Let  $F$  be a free group of finite rank  $q$ . Then, the group of all automorphisms of  $F$ ,  $Aut(F)$ , is generated by the elementary Nielsen transformations  $(T1)$  and  $(T2)$ .

More precisely: Each automorphism of  $F$  is describable as a regular Nielsen transformation between two basis of  $F$ , and, each regular Nielsen transformation between two basis of  $F$  defines an automorphism of  $F$ .

**Proposition 4.2.5.** [CgRR08, Theorem 2.3] or [LS77, Proposition 2.2]

If  $U = \{u_1, u_2, \dots, u_m\}$  is finite, then  $U$  can be carried by a Nielsen transformation into some  $V$ , such that  $V$  is Nielsen reduced. It is  $|V| = rank(\langle V \rangle) \leq m$ .

**Proposition 4.2.6.** [MKS66, Corollary 3.1]

Let  $H$  be a finitely generated subgroup of the free group  $F$  on the free generating set  $X$ . Let  $U = \{u_1, u_2, \dots, u_t\}$ ,  $u_i$  words in  $X$ , be a Nielsen reduced set, which generates  $H$ . Then, out of all systems of generators for  $H$ , the set  $U$  has the shortest total  $X$ -length, which is  $\sum_{i=1}^t |u_i|_X$ .

**Remark 4.2.7.** If  $F_V$  is a finitely generated subgroup of  $F = \langle X \mid \quad \rangle$ , with free generating set  $V = \{v_1, v_2, \dots, v_N\}$ ,  $v_i$  words in  $X$ , then there exist only finitely many Nielsen reduced sets  $U_i = \{u_{i1}, u_{i2}, \dots, u_{iN}\}$ ,  $i = 1, 2, \dots, \ell$ , to  $V$ , which are Nielsen equivalent. With the help of a (lexicographical) order relation  $\prec$  the smallest set  $U_s$ , in the set of all Nielsen reduced sets  $U_{Nred}^V := \{U_1, U_2, \dots, U_\ell\}$  to  $V$ , can be uniquely marked. With the use of a regular Nielsen transformation it is possible to obtain this marked set  $U_s$  starting from any arbitrary set in  $U_{Nred}^V$ .

**Example 4.2.8.** Let  $F$  be a free group on the free generating set  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ . For example (see [MKS66]) we can define a (lexicographical) order relation  $\prec$  among words  $w \in F$  as follows:

If  $|w| < |u|$  for words  $w, u \in F$ , then  $w \prec u$ .

It is

$$x_1 \prec x_1^{-1} \prec x_2 \prec x_2^{-1} \prec \dots \prec x_q \prec x_q^{-1}.$$

If  $|w| = |u|$  for words  $w, u \in F$  and they first differ in their  $k$ th term, then order  $w$  and  $u$  according to their  $k$ th term.

For example

$$1 \prec x_2 \prec x_1x_q \prec x_1x_q^{-1} \prec x_2^3.$$

Let  $F$  be a free group on the free generating set  $X = \{x, y\}$ . The sets  $\{y^2, yxy^{-1}\}$  and  $\{y^2, y^{-1}xy\}$  are Nielsen reduced (see Example 4.2.16). It is

$$y^2 \prec yxy^{-1} \prec y^{-1}xy,$$

thus, here  $\{y^2, yxy^{-1}\}$  is the smallest set of  $\{\{y^2, yxy^{-1}\}, \{y^2, y^{-1}xy\}\}$  concerning this (lexicographical) order relation  $\prec$ .

**Corollary 4.2.9.** [CgRR08, Korollar 2.9]

Let  $F$  be a free group with basis  $X$  and let  $U$  be a subset of  $X$  which is Nielsen reduced. Then it is

$$X^{\pm 1} \cap \langle U \rangle = X^{\pm 1} \cap U^{\pm 1}.$$

Especially, if  $U$  is also a basis for  $F$ , then  $X^{\pm 1} = U^{\pm 1}$ .

**Corollary 4.2.10.** [CgRR08, Korollar 2.4]

Let  $F$  be a free group with basis  $X$  and let  $U$  be a subset of  $X$  which is Nielsen reduced. Then there exists for each  $u \in U^{\pm 1}$  words  $\ell(u)$ ,  $m(u)$  and  $r(u)$  with  $m(u) \neq 1$ , such that

1.  $u \equiv \ell(u)m(u)r(u)$ ,
2. if  $w = u_1u_2 \cdots u_t$ ,  $t \geq 0$ ,  $u_i \in U^{\pm 1}$ ,  $u_iu_{i+1} \neq 1$  for all  $i = 1, 2, \dots, t-1$ , then the elements  $m(u_1), m(u_2), \dots, m(u_t)$  stay unreduced in the reduced form of  $w$ .

Especially, it is  $|w|_X = |u_1u_2 \cdots u_{i-1}\ell(u_i)|_X + |m(u_i)|_X + |r(u_i)u_{i+1}u_{i+2} \cdots u_t|_X$  for all  $1 \leq i \leq t$ .

**Remark 4.2.11.** Because of Corollary 4.2.10, there exists for each  $u \in U^{\pm 1}$  a **stable letter**  $x_j^{\pm 1} \in m(u)$ . This stable letter is in each freely reduced word between elements of  $U$  at the place where  $u$  appears. For the element  $u^{-1}$  the stable letter is the inverse of the stable letter of  $u$  (if there is a reduction in  $uv$ , there is also a reduction in  $v^{-1}u^{-1}$ ). It is also possible that  $u$  has more than one stable letter.

**Corollary 4.2.12.** [CgRR08, Korollar 2.5]

Let  $F$  be a free group with basis  $X$  and let  $U$  be a subset of  $X$  which is Nielsen reduced. Let  $w = u_1u_2 \cdots u_k$ ,  $k \geq 0$ ,  $u_i \in U^{\pm 1}$ ,  $u_iu_{i+1} \neq 1$  for  $1 \leq i < k$ . Then

- (a)  $|w|_X \geq k$ ,
- (b)  $|w|_X \geq \frac{1}{2}|u_1|_X + \frac{1}{2}|u_k|_X + k - 2$  for  $k \geq 2$ ,
- (c)  $|w|_X \geq |u_i|_X$  for all  $1 \leq i \leq k$ .

**Theorem 4.2.13.** [CgRR08, Satz 2.6]

Let  $U$  be Nielsen reduced, then  $\langle U \rangle$  is free on  $U$ .

**Theorem 4.2.14.** (Nielsen-Schreier) [CgRR08, Satz 2.11]

*Every finitely generated subgroup of a free group is free.*

For the next lemma we need some notations. Let  $w \neq 1$  be a freely reduced word in  $X$ . The initial segment  $s$  of  $w$  which is “a little more than half” of  $w$ , that is,  $\frac{1}{2}|w| < |s| \leq \frac{1}{2}|w| + 1$ , is called the **major initial segment** of  $w$ . The **minor initial segment** of  $w$  is that initial segment  $s'$  which is “a little less than half” of  $w$ , that is,  $\frac{1}{2}|w| - 1 \leq |s'| < \frac{1}{2}|w|$ . Similarly, **major** and **minor terminal segments** are defined.

If the free length of the word  $w$  is even, we call the initial segment  $s$  of  $w$ , with  $|s| = \frac{1}{2}|w|$  the **left half** of  $w$ . Analogously, we call the terminal segment  $s'$  of  $w$ , with  $|s'| = \frac{1}{2}|w|$  the **right half** of  $w$ .

Let  $\{w_1, w_2, \dots, w_m\}$  be a set of freely reduced words in  $X$ , which are not the identity. An initial segment of a  $w$ -symbol (that is, of either  $w_i$  or  $w_i^{-1}$ , which are different  $w$ -symbols) is called **isolated** if it does not occur as an initial segment of any other  $w$ -symbol. Similarly, a terminal segment is isolated if it is a terminal segment of a unique  $w$ -symbol.

**Lemma 4.2.15.** [MKS66, Lemma 3.1]

*Let  $M = \{w_1, w_2, \dots, w_m\}$  be a set of freely reduced words in  $X$  with  $w_j \neq 1$ ,  $1 \leq j \leq m$ . Then  $M$  is Nielsen reduced if and only if the following conditions are satisfied:*

1. *Both the major initial and major terminal segments of each  $w_i \in M$  are isolated.*
2. *For each  $w_i \in M$  of even free length, either its left half or its right half is isolated.*

**Example 4.2.16.** Let  $F$  be a free group of rank 2 and let  $X = \{x, y\}$  be its free generating set.

1. The set  $U_1 = \{y^2, yxy^{-1}\}$  is Nielsen reduced. To prove this, we first prove the condition 1. of Lemma 4.2.15: The major initial and major terminal segments are listed in Table 4.1 (page 99).

Table 4.1.: Major initial and major terminal segment for elements in  $U_1$

Element in $U_1$	Major initial segment	Major terminal segment
$y^2$	$y^2$	$y^2$
$yxy^{-1}$	$yx$	$xy^{-1}$

The major initial segment  $y^2$  is no initial segment of  $yxy^{-1}$  and  $yx^{-1}y^{-1}$ . The major initial segment  $yx$  is no initial segment of  $y^2$  and  $y^{-2}$ . The major terminal segment  $y^2$  is no terminal segment of  $yxy^{-1}$  and  $yx^{-1}y^{-1}$ . The major terminal segment  $xy^{-1}$  is no terminal segment of  $y^2$  and  $y^{-2}$ .

Hence, the major initial and major terminal segment of each element in  $U_1$  is isolated. Therefore, condition 1. of Lemma 4.2.15 holds.

Second, we prove the condition 2. of Lemma 4.2.15: The left and the right half of the elements with even length are listed in Table 4.2 (page 100).

Table 4.2.: Left and right half of elements in  $U_1$  of even length

Element in $U$	Left half	Right half
$y^2$	$y$	$y$

The left half  $y$  is an initial segment of  $xyy^{-1}$  and  $yx^{-1}y^{-1}$  but the right half  $y$  is no terminal segment of  $xyy^{-1}$  and  $yx^{-1}y^{-1}$ . Hence, condition 2. of Lemma 4.2.15 holds. Therefore, we proved that  $U_1 = \{y^2, xyx^{-1}\}$  is Nielsen reduced.

2. The set  $U_2 = \{y^2, y^{-1}xy\}$  is Nielsen reduced. We first prove the condition 1. of Lemma 4.2.15: The major initial and major terminal segments are listed in Table 4.3 (page 100).

Table 4.3.: Major initial and major terminal segment for elements in  $U_2$

Element in $U_2$	Major initial segment	Major terminal segment
$y^2$	$y^2$	$y^2$
$y^{-1}xy$	$y^{-1}x$	$xy$

The major initial segment  $y^2$  is no initial segment of  $y^{-1}xy$  and  $y^{-1}x^{-1}y$ . The major initial segment  $y^{-1}x$  is no initial segment of  $y^2$  and  $y^{-2}$ . The major terminal segment  $y^2$  is no terminal segment of  $y^{-1}xy$  and  $y^{-1}x^{-1}y$ . The major terminal segment  $xy$  is no terminal segment of  $y^2$  and  $y^{-2}$ .

Hence, the major initial and major terminal segment of each element in  $U_2$  is isolated. Therefore, condition 1. of Lemma 4.2.15 holds.

Second, we prove the condition 2. of Lemma 4.2.15: The left and the right half of the elements with even length are listed in Table 4.4 (page 100).

Table 4.4.: Left and right half of elements in  $U_2$  of even length

Element in $U_2$	Left half	Right half
$y^2$	$y$	$y$

The right half  $y$  is a terminal segment of  $y^{-1}xy$  and  $y^{-1}x^{-1}y$  but the left half  $y$  is no initial segment of  $y^{-1}xy$  and  $y^{-1}x^{-1}y$ . Hence, condition 2. of Lemma 4.2.15 holds. Therefore, we proved that  $U_2 = \{y^2, y^{-1}xy\}$  is Nielsen reduced.

We now have two sets  $U_1$  and  $U_2$  which are Nielsen reduced and it is  $U_1 \neq U_2$ , but the sets  $U_1$  and  $U_2$  are Nielsen equivalent and hence generate the same group.

It is

$$\begin{aligned}
 U_1 = \{y^2, yxy^{-1}\} \text{ we get } (y^2, yxy^{-1}) &\xrightarrow{(N2)_{2,1}} (y^2, yxy) \\
 &\xrightarrow{(N1)_2} (y^2, y^{-1}x^{-1}y^{-1}) \\
 &\xrightarrow{(N2)_{2,1}} (y^2, y^{-1}x^{-1}y) \\
 &\xrightarrow{(N1)_2} (y^2, y^{-1}xy) \text{ hence } U_2 = \{y^2, y^{-1}xy\}.
 \end{aligned}$$

The set  $U_3 := \{y^2, yxy\}$  generates also the same group as  $U_1$  and  $U_2$ , see the above Nielsen transformation from  $U_1$  to  $U_2$ ; but  $U_3$  is not Nielsen reduced, because condition 2. of Lemma 4.2.15 does not hold for  $U_3$ . To prove this, the left and the right half of the elements with even length in  $U_3$  are listed in Table 4.5 (page 101).

Table 4.5.: Left and right half of elements in  $U_3$  of even length

Element in $U_3$	Left half	Right half
$y^2$	$y$	$y$

The left half  $y$  is an initial segment of  $yxy$  and the right half  $y$  is a terminal segment of  $yxy$ . Hence, condition 2. of Lemma 4.2.15 holds not and  $U_3$  is not Nielsen equivalent.

**Remark 4.2.17.** In [Ste89] an algorithm, using elementary Nielsen transformations, is presented which, given a finite set  $S$  of  $m$  words of a free group, returns a set  $S'$  of Nielsen reduced words, such that  $\langle S \rangle = \langle S' \rangle$ ; the algorithm runs in  $\mathcal{O}(\ell^2 m^2)$ , where  $\ell$  is the maximum free length of a word in  $S$ .

For some cryptographic protocols in the next chapters we would like to use a faithful representation from a free group  $F$  with free generating set  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ , into the special linear group  $\text{SL}(2, \mathbb{Q})$ . Therefore, we could use the following theorem.

**Theorem 4.2.18.** [Leh64]

Let  $F$  be a free group with countable number of generators  $x_1, x_2, \dots$ ; corresponding to  $x_j$  define

$$M_j = \begin{pmatrix} -r_j & -1 + r_j^2 \\ 1 & -r_j \end{pmatrix},$$

with  $r_j \in \mathbb{Q}$  and the inequalities

$$r_{j+1} - r_j \geq 3 \quad \text{and} \quad r_1 \geq 2. \tag{4.1}$$

Then  $G^*$  generated by  $\{M_1, M_2, \dots\}$  is isomorphic to  $F$ .

**Definition 4.2.19.** [CgRR08]

Let  $F$  be a free group of rank  $q$  and let  $G$  be a free subgroup of  $F$  with rank  $m$ . An element  $g \in G$  is called a **primitive element** of  $G$ , if a basis  $U$  of  $G$  with  $g \in U$  exists.

**Definition 4.2.20.** [Rot95]

A subgroup  $H$  of  $F$  is called **characteristic** in  $F$  if  $\varphi(H) = H$  for every automorphism  $\varphi$  of  $F$ .

**Proposition 4.2.21.** [MS03]

The number of primitive elements of free length  $k$  of the free group  $F = \langle x_1, x_2 \mid \ \rangle$  (and therefore, in any free group  $F = \langle x_1, x_2, \dots, x_q \mid \ \rangle$ ,  $q \geq 2$ ) is:

1. more than  $\frac{8}{3\sqrt{3}} \cdot (\sqrt{3})^k$  if  $k$  is odd;
2. more than  $\frac{4}{3} \cdot (\sqrt{3})^k$  if  $k$  is even.

**Theorem 4.2.22.** [BMS02]

If  $P(q, k)$  is the number of primitive elements of free length  $k$  of the free group

$$F = \langle x_1, x_2, \dots, x_q \mid \ \rangle,$$

$q \geq 3$ , then for some constants  $c_1$  and  $c_2$ , we have

$$c_1 \cdot (2q - 3)^k \leq P(q, k) \leq c_2 \cdot (2q - 2)^k.$$

**Definition 4.2.23.** [FGMRS14, Definition 3.2.1.]

Let  $G = \langle X \mid R \rangle$  be a group presentation. We form a graph  $\Gamma(G, X)$  in the following way. Let  $X^{\pm 1} = X \cup X^{-1}$ . For the vertex set of  $\Gamma(G, X)$  we take the elements of  $G$ , that is,  $V(\Gamma) = \{g \mid g \in G\}$ . The edges of  $\Gamma$  are given by the set  $E(\Gamma) = \{(g, x) \mid g \in G, x \in X^{\pm 1}\}$ . We call  $g$  the initial point and  $gx$  the terminal point. Two points  $g_1$  and  $g_2$  in the vertex set are connected by an edge if  $g_2 = g_1x$  for some  $x \in X^{\pm 1}$ . We have  $(g, x)^{-1} = (gx, x^{-1})$ . This gives an oriented graph called the **Cayley graph** on  $G$  on the generating set  $X$ .

We call  $x$  the label on the edge  $(g, x)$ . Given a  $g \in G$  then  $g$  is represented by at least one word  $w$  in  $X^{\pm 1}$ . This represents a path in the Cayley graph. The length of the word  $w$  is the length of the path. Each closed path in the Cayley graph represents a relator.

**Example 4.2.24.** 1. The Cayley graph for the symmetric group

$$S_3 = \langle x, y \mid x^2 = y^3 = (xy)^2 = 1 \rangle$$

is given in Figure 4.2.

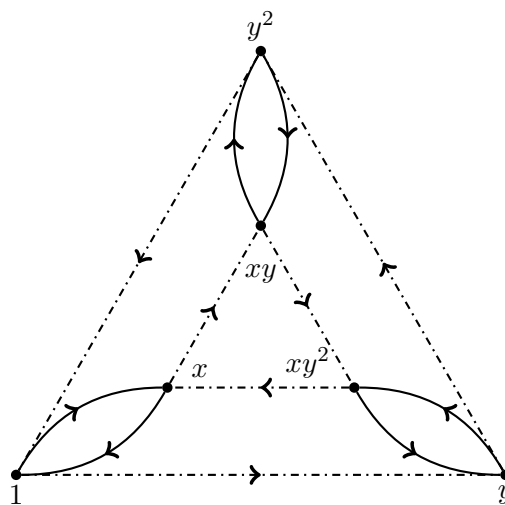


Figure 4.2.: Cayley graph for  $S_3$

2. The Cayley graph for the free group  $F = \langle x, y \mid \ \rangle$  on two symbols is given in Figure 4.3.

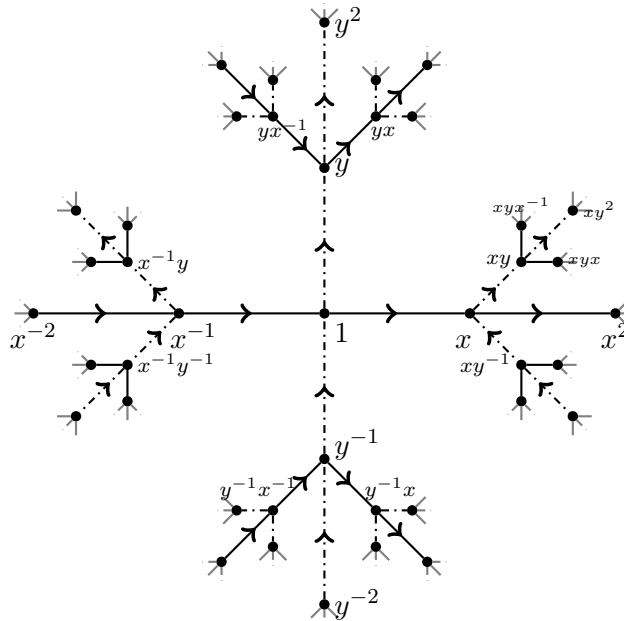


Figure 4.3.: Cayley graph for the free group  $F$  of rank 2

**Remark 4.2.25.** Let  $F$  be a group and  $L$  a natural number. The ball  $B(F, L)$  in the Cayley graph from the group  $F$  contains all elements  $y \in F$  with free length  $|y| \leq L$ .

### 4.3. Fundamental problems in group theory

M. Dehn formalized 1911 in [Deh11] three fundamental problems for groups, see also for instance [MKS66] or [MSU08]. These problems are the word problem, the conjugacy problem and the isomorphism problem.

**Problem 4.3.1.** Word problem:

Let  $G = \langle X \mid R \rangle$  be a presentation of a group and  $g \in G$  a given word in  $X$ . Determine algorithmically (in finitely many steps) if  $g$  represents the identity or not.

**Problem 4.3.2.** Conjugacy problem:

Let  $G = \langle X \mid R \rangle$  be a presentation of a group and two elements  $g, h \in G$  are given. Determine algorithmically (in finitely many steps) if  $g$  is conjugated to  $h$  in  $G$  or not, that means, determine whether or not there is an element  $x \in G$ , such that  $x^{-1}gx = h$ .

**Problem 4.3.3.** Isomorphism problem:

Let  $G = \langle X \mid R \rangle$  and  $H = \langle Y \mid S \rangle$  be two presentations of two group. Determine algorithmically (in finitely many steps) if  $G$  is isomorphic to  $H$  or not.

**Remark 4.3.4.** Problem 4.3.1 is a special case of Problem 4.3.2. If we select  $h$  to be the identity then the solution to the conjugacy problem yields a solution to the word problem.

These problems have not been solved in general. Indeed, they have been solved for some classes of presentations of one specialized form or another, see for example [MKS66, Section 1.3]. However, they are solvable in abstract free groups.

**Corollary 4.3.5.** [CgRR08, Korollar 1.6] Word problem in free groups:

Two words in a free group  $F = \langle X \mid \ \rangle$  are equivalent if and only if they have the same reduced word.

**Corollary 4.3.6.** [CgRR08, Korollar 1.7] Conjugacy problem in free groups:

Two reduced words  $w_1$  and  $w_2$  are conjugated in a free group  $F = \langle X \mid \ \rangle$  if and only if they are of the form

$$w_1 = hkgh^{-1} \quad \text{and} \quad w_2 = lgkl^{-1},$$

whereby  $kg$  and  $gk$ , respectively, do not end with the inverse of their first letter.

**Theorem 4.3.7.** [CgRR08, Satz 1.9] Isomorphism problem in free groups:

Let  $X$  and  $Y$  be two sets. Let  $G = \langle X \mid \ \rangle$  and  $H = \langle Y \mid \ \rangle$  be two free groups on  $X$  and  $Y$ , respectively. The free group  $G$  is isomorphic to the free group  $H$  if and only if  $|X| = |Y|$ .

A further problem, which is a more general problem than the word problem and is needed for some of the developed cryptographic protocols based on combinatorial group theory, is the membership problem or also called extended word problem.

**Problem 4.3.8.** Membership problem:

Given a recursively presented group  $G$ , a subgroup  $H$  of  $G$  generated by  $h_1, h_2, \dots, h_k$  and an element  $g \in G$ , determine whether or not  $g \in H$ .

A related problem (to the membership problem) is the constructive membership problem.

**Problem 4.3.9.** Constructive membership problem:

Given a recursively presented group  $G$ , a subgroup  $H$  of  $G$  generated by  $h_1, h_2, \dots, h_k$  and an element  $h \in H$ , find an expression of  $h$  in terms of  $h_1, h_2, \dots, h_k$ .

The membership problem is solvable for abstract free groups. Let  $F$  be a free group and let  $w \in F$ . Recall, we write  $w \equiv w_1 w_2 \cdots w_k$  if there are no cancellations between the words  $w_i$  and  $w_{i+1}$ ,  $1 \leq i < k$ , that means  $|w| = |w_1| + |w_2| + \cdots + |w_k|$ .

**Theorem 4.3.10.** [CgRR08, Satz 2.21] Membership problem in free groups:

Let  $F$  be a free group with basis  $X$ ,  $G = \langle g_1, g_2, \dots, g_n \rangle$  a finitely generated subgroup of  $F$ ,  $g_i$  words in  $X$ , and  $w \in F$ . The following instruction defines an algorithm to decide if  $w \in G$ :

1. Calculate to the set  $\{g_1, g_2, \dots, g_n\}$  a Nielsen equivalent set  $U$ , which is Nielsen reduced.
2. Write each  $u \in U$  in the form  $u \equiv \ell(u)m(u)r(u)$  as in Corollary 4.2.10 with a stable part  $m(u)$ .
3. If  $w = 1$ , then the out put is “ $w \in G$ ” and stop.
4. If there exists an element  $u_0 \in U$  with  $w \equiv \ell(u_0)m(u_0)w'$  for a  $w' \in F$ , go to step 5. If there exists an element  $u_0 \in U$  with  $w \equiv r(u_0)^{-1}m(u_0)^{-1}w'$  for a  $w' \in F$ , go to step 6. Otherwise the output is “ $w \notin G$ ” and stop.
5. Replace  $w$  by  $u_0^{-1}w$  and go to step 3.
6. Replace  $w$  by  $u_0w$  and go to step 3.

**Remark 4.3.11.** If the used elements of the set  $U$  in step 5. and step 6. are noted down, it is possible to write the word  $w$  as a product of elements in  $U^{\pm 1}$  if  $w \in G$ . Thus, with Theorem 4.3.10, the constructive membership problem is solvable in free groups  $F$  for words  $w \in F$  and subgroups  $G = \langle U \rangle$  of  $F$  with a Nielsen reduced set  $U$ .



**Remark 4.3.12.** In [CgRR08] it is shown that

$$U = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{und} \quad T = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

generate the special linear group  $\text{SL}(2, \mathbb{Z}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid a, b, c, d \in \mathbb{Z} \text{ and } ab - cd = 1 \right\}$ . The proof gives an algorithm how to write each element  $A \in \text{SL}(2, \mathbb{Z})$  in terms of  $U$  and  $T$  and hence solves the constructive membership problem in this situation. We now take a look at this proof.

Let  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \text{SL}(2, \mathbb{Z})$ . We consider the case where  $c \neq 0$ .

Case 1:  $|a| < |c|$ .

Calculate

$$TA = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} -c & -d \\ a & b \end{pmatrix} =: \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix}.$$

Therefore, we get the case 2 with  $|a'| > |c'|$ .

Case 2:  $|a| > |c|$ .

Then, there are  $q, r \in \mathbb{N}$  with  $|a| = q \cdot |c| + r$  and  $0 \leq r < |c|$ . It follows  $a = p \cdot c + s$  with  $p = \pm q$ ,  $s = \pm r$  and  $0 \leq |s| < |c|$ . Calculate

$$U^{-p}A = \begin{pmatrix} 1 & -p \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} s & b - pd \\ c & d \end{pmatrix} =: \begin{pmatrix} a'' & b'' \\ c'' & d'' \end{pmatrix}.$$

Now,  $|a''| < |c''|$  and we are in case 1. Therefore, together with case 1 and case 2 we end up in finitely many steps with  $s = 0$  (because this is Euclid's algorithm) and thus with case 1 we get a matrix  $SA = \begin{pmatrix} \alpha & \beta \\ 0 & \delta \end{pmatrix} \in \text{SL}(2, \mathbb{Z})$  with  $S \in \langle T, U \rangle$ .

Case 3:  $|a| = |c|$ . We consider two cases

i)  $a = c$ :

Calculate

$$TUTA = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} -a & -b \\ 0 & b - d \end{pmatrix}.$$

ii)  $a = -c$ :

Calculate

$$T^{-1}UA = \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} c & d \\ 0 & -b - d \end{pmatrix}.$$

Hence, we get also a matrix  $SA = \begin{pmatrix} \alpha & \beta \\ 0 & \delta \end{pmatrix} \in \text{SL}(2, \mathbb{Z})$  with  $S \in \langle T, U \rangle$ .

Thus, we can reduce the case  $c \neq 0$  to the case  $c = 0$  and consider the matrix

$$SA = \begin{pmatrix} \alpha & \beta \\ 0 & \delta \end{pmatrix} \in \text{SL}(2, \mathbb{Z}),$$

with  $S \in \langle T, U \rangle$ . Due to  $\alpha \cdot \delta = 1$  it follows  $\alpha = \delta = \pm 1$ . Without loss of generality let  $\alpha = \delta = 1$  (otherwise calculate  $T^2SA$ , because  $T^2 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ ). Therefore,  $SA = U^\beta$  and thus  $A = S^{-1}U^\beta \in \langle T, U \rangle$ . If the used matrixes for  $S$  and  $U^\beta$  are stored we are able to write

the matrix  $A$  in terms of  $U$  and  $T$  and hence solve the constructive membership problem in this situation.

This can also be used to show, that the constructive membership problem is solvable for the modular group  $\Gamma = \left\{ z \mapsto \frac{az+b}{cz+d} \mid z \in \mathbb{C} \cup \{\infty\} \mid a, b, c, d \in \mathbb{Z} \text{ and } ad - bc = 1 \right\}$ . It is

$$\Gamma \cong \text{SL}(2, \mathbb{Z}) / \{\pm E_2\} = \text{PSL}(2, \mathbb{Z}), \quad \text{with } E_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The group  $\Gamma$  is generated by the transformations  $\bar{U} : z \mapsto z + 1$  and  $\bar{T} : z \mapsto -\frac{1}{z}$ , because  $U$  and  $T$  generate  $\text{SL}(2, \mathbb{Z})$ . From  $\Gamma = \langle \bar{T}, \bar{U} \rangle$  it follows that also  $\Gamma = \langle \bar{R}, \bar{T} \rangle$ , with  $\bar{R} = \bar{T}\bar{U}$ , and it is

$$\Gamma = \langle \bar{T}, \bar{R} \mid \bar{T}^2 = \bar{R}^3 = 1 \rangle.$$

This can be seen as follow:

It is  $\bar{R} : z \mapsto -\frac{1}{z+1}$ ;  $\bar{R}^2 : z \mapsto -\frac{z+1}{z}$  and  $\bar{R}^3 : z \mapsto z$  thus  $\bar{R}$  is of order 3.

Let  $\mathbb{R}^- = \{x \in \mathbb{R} \mid x < 0\}$  and  $\mathbb{R}^+ = \{x \in \mathbb{R} \mid x > 0\}$ , thus it is  $\bar{T}(\mathbb{R}^-) \subset \mathbb{R}^+$  and  $\bar{R}^\alpha(\mathbb{R}^+) \subset \mathbb{R}^-$  for  $\alpha = 1, 2$ . The relations  $\bar{T}^2 = \bar{R}^3 = 1$  are defining relations for  $\Gamma$ , because: Let  $\bar{S} \in \Gamma$ , after applying the relations  $\bar{T}^2 = \bar{R}^3 = 1$  and suitable conjugations, we can assume that  $\bar{S} = 1$  or

$$\bar{S} = \bar{R}^{\alpha_1} \cdot \bar{T} \cdot \bar{R}^{\alpha_2} \dots \bar{T} \cdot \bar{R}^{\alpha_{n+1}},$$

with  $1 \leq \alpha_i \leq 2$  and  $\alpha_1 = \alpha_{n+1}$ . Let  $x \in \mathbb{R}^+$  for the second case, then it is  $\bar{S}(x) \in \mathbb{R}^-$  and therefore  $\bar{S} \neq 1$ .

**Theorem 4.3.13.** [GS07, Theorem 1.3]

*The membership problem for the modular group, that is,  $\text{PSL}(2, \mathbb{Z}) = \text{SL}(2, \mathbb{Z}) / \{\pm E_2\}$ , with  $E_2$  the identity matrix in  $\text{SL}(2, \mathbb{Z})$ , is decidable in polynomial time.*

**Remark 4.3.14.** There is no algorithm known to solve the (constructive) membership problem for (discrete) free subgroups of  $\text{SL}(2, \mathbb{Q})$  of rank greater than or equal to 2. B. Eick, M. Kirschmer and C. Leedham-Green present in their paper [EKL14] a practical algorithm to solve the constructive membership problem for discrete free subgroups of rank 2 of  $\text{SL}(2, \mathbb{R})$ . For example, the subgroup  $\text{SL}(2, \mathbb{Z})$  of  $\text{SL}(2, \mathbb{R})$  is discrete, but they also mention, that it is an open problem to solve the membership problem for (discrete) free subgroups of  $\text{SL}(2, \mathbb{R})$  with arbitrary rank  $m \geq 2$ .

**Remark 4.3.15.** The group  $\text{SL}(2, \mathbb{R})$  is a topological group with respect to the metric  $d$  on  $\text{SL}(2, \mathbb{R})$  defined by  $d(M, N) = \|M - N\|$ , where  $\|M\| = \sqrt{\text{tr}(MM^t)}$ . A subgroup  $G$  of  $\text{SL}(2, \mathbb{R})$  is said to be **discrete** if  $G$  is discrete with respect to this topology. In other words, a subgroup  $G$  of  $\text{SL}(2, \mathbb{R})$  is discrete if  $\inf\{\|M - I\| \mid M \in G, M \neq \pm I\} \neq 0$ , see for instance [Bea83].

## 4.4. Whitehead-Automorphisms

The cryptographic protocols, **Protocol 6** to **Protocol 12**, which we explain in the next chapters, are based on automorphisms of  $F$ , with  $F = \langle X \mid \ \rangle$  a free group on free generating set  $X$  with  $|X| = q \geq 2$ . A fixed set of randomly chosen automorphisms is part of the key space for the private key cryptosystem. These automorphisms should be chosen randomly. It is known, see Proposition 4.2.4, that the Nielsen transformations generate the automorphism group  $\text{Aut}(F)$ .

For a realization of a random choice procedure the Whitehead-Automorphisms will be used. The following approach for choosing automorphisms randomly of  $Aut(F)$  is published in [MR16].

**Definition 4.4.1.** Whitehead-Automorphisms

1. Invert the letter  $a$  and leave all other letters invariant:

$$i_a(b) = \begin{cases} a^{-1} & \text{for } a = b \\ b & \text{for } b \in X \setminus \{a\}. \end{cases}$$

There are  $q$  **Whitehead-Automorphisms** of this type.

2. Let  $a \in X$  and  $L, R, M$  be three pairwise disjoint subsets of  $X$ , with  $a \in M$ . Then the tuple  $(a, L, R, M)$  defines a **Whitehead-Automorphism**  $W_{(a,L,R,M)}$  as follows

$$W_{(a,L,R,M)}(b) = \begin{cases} ab & \text{for } b \in L \\ ba^{-1} & \text{for } b \in R \\ aba^{-1} & \text{for } b \in M \\ b & \text{for } b \in X \setminus (L \cup M \cup R). \end{cases}$$

There are  $q \cdot 4^{q-1}$  automorphisms of this type.

Note, that  $W_{(a,L,R,M)}^{-1} = i_a \circ W_{(a,L,R,M)} \circ i_a$ .

With this definition it is clear how the Whitehead-Automorphisms can be generated as a product of regular Nielsen transformations. Conversely, the Whitehead-Automorphisms generate the group of the Nielsen transformations and therefore also the automorphism group  $Aut(F)$  (see also [DKR13]). With the Whitehead-Automorphisms it is simple to realize a random choice of automorphisms. We now give an approach for this choice.

**An approach for choosing automorphisms randomly of  $Aut(F)$ :**

Let  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ , be the free generating set for the free group  $F$ .

1. First of all it should be decided in which order an automorphism  $f_i$  is generated by automorphisms of type  $i_a$  and  $W_{(a,L,R,M)}$ . For this purpose an automorphism of type  $i_a$  is identified with a zero and  $W_{(a,L,R,M)}$  with a one. A sequence of zeros and ones is randomly generated. This sequence is translated to randomly chosen Whitehead-Automorphisms and hence presents an automorphism  $f_i \in Aut(F)$ . This translation is as follows:
  - 2.1. For a zero in the sequence we generate  $i_a$  randomly: choose a random number  $z$ , with  $1 \leq z \leq q$ ; hence an element  $a \in X$  must be chosen to declare the automorphism. Then it is  $a := x_z$  and hence  $x_z$  is replaced by  $x_z^{-1}$  and all other letters are invariant.
  - 2.2. For a one in the sequence we generate  $W_{(a,L,R,M)}$  randomly: choose a random number  $z$ , with  $1 \leq z \leq q$ . Hence, it is  $a := x_z$ . Moreover, it is  $a \in M$ . After this the disjoint sets  $L, R, M \subseteq X$  are chosen randomly. One possible approach is the following:
    - a) Choose random numbers  $z_1, z_2$  and  $z_3$  with

$$\begin{aligned} 0 &\leq z_1 \leq q - 1, \\ 0 &\leq z_2 \leq q - 1 - z_1, \\ 0 &\leq z_3 \leq q - 1 - z_1 - z_2. \end{aligned}$$

If we are in the situation of  $z_1 = z_2 = z_3 = 0$  we get the identity  $id_X$ . If this case arises a random number  $\tilde{z}$  from the set  $\{1, 2, \dots, q\} \setminus \{z\}$  is chosen and hence the element  $x_{\tilde{z}}$  is assigned randomly to one of the sets  $L$ ,  $R$  or  $M$ ; therefore the identity is avoided.

It is

$$|L| = z_1, \quad |R| = z_2, \quad |M| = z_3 + 1.$$

b) Choose  $z_1$  pairwise different random numbers  $r_1, r_2, \dots, r_{z_1}$  of the set

$$\{1, 2, \dots, q\} \setminus \{z\}.$$

Then  $L$  is the set

$$L = \{x_{r_1}, x_{r_2}, \dots, x_{r_{z_1}}\}.$$

c) Choose  $z_2$  pairwise different random numbers  $p_1, p_2, \dots, p_{z_2}$  of the set

$$\{1, 2, \dots, q\} \setminus (\{z\} \cup \{r_1, r_2, \dots, r_{z_1}\}).$$

Then  $R$  is the set

$$R = \{x_{p_1}, x_{p_2}, \dots, x_{p_{z_2}}\}.$$

d) Choose  $z_3$  pairwise different random numbers  $t_1, t_2, \dots, t_{z_3}$  of the set

$$\{1, 2, \dots, q\} \setminus (\{z\} \cup \{r_1, r_2, \dots, r_{z_1}\} \cup \{p_1, p_2, \dots, p_{z_2}\}).$$

Then  $M$  is the set

$$M = \{x_{t_1}, x_{t_2}, \dots, x_{t_{z_3}}\} \cup \{a\}.$$

**Remark 4.4.2.** If Alice and Bob use Whitehead-Automorphisms to generate automorphisms on a free group with free generating set  $X$  they should take care, that there are no sequences of the form

1.  $i_a \circ i_a = id_X$ ,
2.  $W_{(a,L,R,M)} \circ i_a \circ \underbrace{W_{(a,L,R,M)}}_{=W_{(a,L,R,M)}^{-1}} \circ i_a = id_X$  or  
 $\underbrace{i_a \circ W_{(a,L,R,M)} \circ i_a}_{=W_{(a,L,R,M)}^{-1}} \circ W_{(a,L,R,M)} = id_X$ ,

for the automorphisms  $f_j$ . They also should not use Whitehead-Automorphisms sequences for  $f_j$ , which cancel each other and so be vacuous for the encryption.

# Chapter 5

## Secret sharing protocols

This chapter introduces **Protocol 5**, which is a purely combinatorial  $(n, t)$ -secret sharing scheme. It uses the combinatorial share distribution method, which D. Panagopoulos describes in [Pan10] for his combinatorial group theoretical  $(n, t)$ -secret sharing scheme.

We realize that this share distribution method is also given as a special case by M. Ito, A. Saito and T. Nishizeki in [ISN87]. We show that if the method in [ISN87] is used to generate a  $(n, t)$ -secret sharing scheme then the same share distribution method as by D. Panagopoulos is described. M. Ito, A. Saito and T. Nishizeki use a multiple assignment scheme, which is a method to distribute to each participant more than only one share, together with a  $(m, m)$ -secret sharing scheme. Thus, we see that the share distribution method by D. Panagopoulos is a special case of paper [ISN87], see Table 5.2 (page 123).

In addition we realize that the purely combinatorial secret sharing scheme (**Protocol 5**) is very similar to a scheme, which J. Benaloh and J. Leichter obtain if they realize a  $(n, t)$ -secret sharing scheme using minimal CNF form, described in their paper [BL90]. We explain this in detail and a summary is given in Table 5.3 (page 129).

**Protocol 5** is published in the survey article [CFMRZ16] as research in the field of secret sharing schemes. It is also published in [MR15].

We start the chapter with a definition of  $(n, t)$ -secret sharing schemes and briefly explain the two first mathematical  $(n, t)$ -secret sharing schemes. One was given by A. Shamir in [Sha79] and the other by G. Blakley in [Bla79]. A. Shamir's secret sharing protocol has become the standard method for solving the  $(n, t)$ -secret sharing problem. He lists some useful properties for  $(n, t)$ -secret sharing schemes in his paper [Sha79], which we also use to analyze different secret sharing schemes and compare them to A. Shamir's scheme.

We explain D. Panagopoulos'  $(n, t)$ -secret sharing scheme and the purely combinatorial  $(n, t)$ -secret sharing scheme (**Protocol 5**). It follows a section about access structures of generalized secret sharing schemes, because the papers [ISN87] and [BL90] examine such structures. Generalized secret sharing schemes realize not only the situation where arbitrary  $t$  of  $n$  persons should be able to reconstruct a secret ( $(n, t)$ -secret sharing scheme) but also some more special structures. For example we assume that in a company with two directors and three vice-directors a secret should be reconstructible if two directors or three vice-directors or one director and two vice-directors of the company cooperate, see Example 5.3.12. We also show the connection of the method by M. Ito, A. Saito and T. Nishizeki to **Protocol 5**, and the connection of the method by J. Benaloh and J. Leichter to **Protocol 5**.

We close this chapter comparing the CFRZ-scheme, D. Panagopoulos' scheme and the purely combinatorial  $(n, t)$ -secret sharing scheme to Shamir's scheme.

**Definition 5.0.1.** Let  $P = \{p_1, p_2, \dots, p_n\}$  be a set of  $n$  participants (or also called trustees). A  $(n, t)$ -secret sharing scheme (or  $(n, t)$ -threshold scheme), with  $n, t \in \mathbb{N}$  and  $t \leq n$ , is a method to split a secret  $S$  into  $n$  shares  $s_1, s_2, \dots, s_n$  and distribute each share  $s_i$  to one participant  $p_i$ ,  $1 \leq i \leq n$ , in such a way that

1. if arbitrary  $t$  or more participants  $P_j = \{p_{j_1}, p_{j_2}, \dots, p_{j_l}\} \subseteq P$ , with  $t \leq l \leq n$ , come together they are able to reconstruct the secret  $S$  with the help of their shares  $s_{j_1}, s_{j_2}, \dots, s_{j_l}$ ;
2. if arbitrary  $t - 1$  or less participants  $P_k = \{p_{k_1}, p_{k_2}, \dots, p_{k_l}\} \subseteq P$ , with  $1 \leq l \leq t - 1$ , come together they are not able to reconstruct the secret  $S$  with the help of their shares  $s_{k_1}, s_{k_2}, \dots, s_{k_l}$ .

The set of all participants for a secret sharing scheme is also called **access control group**, see for example [BFKR15].

If in a  $(n, t)$ -secret sharing scheme  $t - 1$  or less participants combine their shares and they cannot get any information about the secret  $S$ , then we call it a **perfect  $(n, t)$ -secret sharing scheme**. In other words, a perfect  $(n, t)$ -secret sharing scheme is a scheme in which the knowledge of only  $t - 1$  or less shares prove no advantage (no information about the secret  $S$  whatsoever, in the information-theoretic sense) to an outsider (opponent) who knows no shares (see [MvOV97]).

The number  $t$  is called **threshold** and the person who distributes the shares to the participants is called **dealer**.

Mathematical secret sharing schemes were first formalized in 1979 by A. Shamir [Sha79] and independently by G. Blakley [Bla79] and each of them presented a different  $(n, t)$ -secret sharing scheme in his paper.

A. Shamir motivated his paper [Sha79], in which he introduced his  $(n, t)$ -secret sharing scheme, with the following problem from Liu (see [Liu68]):

**Problem 5.0.2.** Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?

If we ask after the smallest number of locks, we must consider the situation in which five scientists come together. There must be at least one lock for which these five scientists together do not have the key, but for which every other scientist of the residuary six must have one. This is computable with the help of the binomial coefficient  $\binom{n}{t} := \frac{n!}{t!(n-t)!}$ , with  $n, t \in \mathbb{N}_0$  and  $t \leq n$ .

There are  $\binom{n}{t}$  possibilities to choose  $t$  elements of a set of  $n$  elements. Thus, there are  $\binom{11}{5} = 462$  groups of five scientists, hence the cabinet must have at least 462 locks.

If we ask after the smallest number of keys to the locks, we are in the situation that each scientist must hold at least one key for every group of five scientists, of which he is not a member. There are  $\binom{10}{5} = 252$  such groups. Thus, each scientist must carry at least 252 keys.

The idea behind  $(n, t)$ -secret sharing schemes, and hence behind A. Shamir's scheme, is that there is just one lock and one key for this lock, but the key is split into subkeys (the shares for the participants). Each participant gets one share, if the desired number of scientists come together they should be able to reconstruct the key combining their subkeys. If fewer scientists wish to open the cabinet, it should be impossible.

**Remark 5.0.3.** In general, if we consider Problem 5.0.2 from Liu for a  $(n, t)$ -secret sharing scheme, we get  $\binom{n}{t-1}$  locks for the cabinet and  $\binom{n-1}{t-1}$  keys for each scientist, due to the illustration in the situation above with  $n = 11$  and  $t = 6$ .

Thus, we claim that a  $(n, t)$ -secret sharing scheme should have less than  $\binom{n}{t-1}$  "locks" and less than  $\binom{n-1}{t-1}$  "keys" for each participant.

**A. Shamir** uses polynomial interpolation for his  $(n, t)$ -secret sharing scheme. Let  $\mathbb{F}$  be any field and let  $(x_1, y_1), (x_2, y_2), \dots, (x_t, y_t)$  be  $t$  points in  $\mathbb{F}^2$  with pairwise distinct  $x_i$ ,  $1 \leq i \leq t$ . We say a polynomial  $g(x)$  over  $\mathbb{F}$  **interpolates** these points if  $g(x_i) = y_i$ ,  $1 \leq i \leq t$ . A. Shamir's secret sharing scheme is based on the following theorem.

**Theorem 5.0.4.** [Atk89]

Let  $\mathbb{F}$  be any field and let  $x_1, x_2, \dots, x_t$  be  $t$  pairwise distinct elements of  $\mathbb{F}$  and let  $y_1, y_2, \dots, y_t$  be any elements of  $\mathbb{F}$ . Then there exists a unique polynomial of degree less than or equal to  $t - 1$  that interpolates the  $t$  points  $(x_i, y_i)$ ,  $1 \leq i \leq t$ .

A. Shamir's  $(n, t)$ -secret sharing scheme is roughly this: The dealer chooses a field  $\mathbb{F}$ . The secret  $S$  is an element in  $\mathbb{F}$ . The dealer picks a polynomial  $g(x)$  of degree  $t - 1$  with the secret  $S$  as constant term, that is,  $g(x) = S + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}$ ,  $a_i \in \mathbb{F}$  and  $a_{t-1} \neq 0$ . He chooses pairwise distinct elements  $x_1, x_2, \dots, x_n \in \mathbb{F}$ , with  $x_i \neq 0$  for all  $1 \leq i \leq n$ , and distributes to each of the  $n$  participants a point  $(x_i, g(x_i))$  as a share. By Theorem 5.0.4 any  $t$  participants can determine the polynomial  $g(x)$  (for example with Lagrange interpolation, see [Atk89]) and hence recover the secret  $S$ . If less than  $t$  people combine their shares any element in  $\mathbb{F}$  can be the constant term and hence the secret. A. Shamir suggested to use  $\mathbb{F} = \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$  where  $p$  is a large prime number.

**G. Blakley's**  $(n, t)$ -secret sharing scheme is a geometrical one. He uses hyperplanes as shares. The secret  $S$  is a point in a  $t$  dimensional vector space and the  $n$  shares are hyperplanes in the  $t$  dimensional vector space with the property that the only intersection point of any arbitrary  $t$  hyperplanes is the secret  $S$ . If less than  $t$  participants combine their shares they cannot reconstruct the secret, but they get information about the secret point  $S$ , because they know that it lies in the intersection of their hyperplanes, hence it is not perfect. If not the whole point is the secret but only one coordinate of the point  $S$ , then they have no more information about the secret than an outsider.

**Remark 5.0.5.** G. Blakley's scheme is less space efficient, for computer storage, than A. Shamir's and also the distributed shares are larger than the secret, whereas A. Shamir gets shares which are of the same size than the secret (see for instance [Sha79] or [BFKR15]). A. Shamir's secret sharing protocol has become the standard method for solving the  $(n, t)$ -secret sharing problem, although there are modifications for different situations as for example a verifying secret sharing variation, see [BFKR15]. A verifying secret sharing protocol makes certain, that the dealer and the participants behave correctly.

## 5.1. D. Panagopoulos' $(n, t)$ -secret sharing scheme

We now introduce a  $(n, t)$ -secret sharing scheme, which uses an interesting method to distribute shares to the participants. This  $(n, t)$ -secret sharing scheme, presented by D. Panagopoulos in [Pan10], is based on combinatorial group theory, see Section 4.1. More precisely, it is based on group presentations of groups with solvable word problem (Problem 4.3.1). The secret is a binary sequence which is reconstructible if the finite presentation of a group

$$G = \langle x_1, x_2, \dots, x_k \mid r_1 = r_2 = \dots = r_m = 1 \rangle$$

is known. The shares are subsets of the set of defining relations  $R = \{r_1, r_2, \dots, r_m\}$  of the used group  $G$ .

The cryptographic protocol is as follows:

Steps for the dealer:

1. A finitely presented group  $G$  with solvable word problem is chosen, it is

$$G = \langle x_1, x_2, \dots, x_k \mid r_1 = r_2 = \dots = r_m = 1 \rangle,$$

with  $m = \binom{n}{t-1}$  and  $k \geq 2$ .

- Let  $A_1, A_2, \dots, A_m$  be an enumeration of the subsets of  $\{1, 2, \dots, n\}$  with  $t-1$  elements. Define  $n$  subsets  $R_1, R_2, \dots, R_n$  of the set  $\{r_1, r_2, \dots, r_m\}$  with the property

$$r_j \in R_i \iff i \notin A_j \quad \text{for } j = 1, 2, \dots, m \text{ and } i = 1, 2, \dots, n.$$

- The dealer distributes to each of the  $n$  participants one of the sets  $R_1, R_2, \dots, R_n$  as a share. The generating set  $\{x_1, x_2, \dots, x_k\}$  is known by each participant.

The secret is a binary sequence  $a_1 a_2 \dots a_\ell$ . The dealer constructs words  $w_i$ ,  $1 \leq i \leq \ell$ , in the group  $G$ , such that

$$w_i = 1 \text{ in } G \iff a_i = 1 \quad \text{for } 1 \leq i \leq \ell.$$

The word  $w_i$  must involve most of the relations  $r_1, r_2, \dots, r_m$  if  $w_i$  represents the identity in  $G$ . Furthermore, all of the defining relations must be used at some point in the construction of some element  $w_i$  (see [Pan10] for a way to create a word which represents the identity). The dealer sends the words  $w_i$ ,  $1 \leq i \leq \ell$ , to the participants.

**Security 5.1.1.** The security of this share distribution method depends on the following facts. The dealer puts  $r_j$  in the set  $R_i$  if and only if  $i \notin A_j$ , for  $j = 1, 2, \dots, m$  and  $i = 1, 2, \dots, n$ . Each subset  $A_1, A_2, \dots, A_m$  has  $t-1$  elements from  $\{1, 2, \dots, n\}$ , hence in each subset  $A_j$  are  $n-(t-1)$  elements from  $\{1, 2, \dots, n\}$  not contained. Thus, each element  $r_j$  is exactly in  $n-(t-1)$  sets  $R_i$  contained. Therefore, each element  $r_j$ ,  $1 \leq j \leq m$ , is not contained in exactly  $t-1$  sets  $R_i$ . Consequently, if just  $t-1$  or less arbitrary sets of the sets  $R_1, R_2, \dots, R_n$  are combined, then there exist a  $j$ , such that the element  $r_j$  is not included in the union of these sets. Further, each  $r_j$  is in each union of at least  $t$  pairwise different sets  $R_i$ .

**Remark 5.1.2.** Each participant  $p_i$  gets a share  $r_j$  for each subset  $A_j$  of size  $t-1$  in which  $i$  is not an element. There are  $\binom{n-1}{t-1}$  many sets with this property. Hence, each participant has as a share a set of  $r := \binom{n-1}{t-1}$  elements.

**Remark 5.1.3.** This is equivalent to the solution of Liu (see Problem 5.0.2 and Remark 5.0.3). We can see the elements in each set  $R_i$  for one participant  $p_i$  as the number of keys for each participant (which is  $\binom{n-1}{t-1}$ ). The number of locks is the number of relations for the group  $G$ , there are  $m = \binom{n}{t-1}$  relations for  $G$ , if the participants know all relations they can reconstruct the secret, which means they can “open the cabinet”.

Steps for the participants:

If  $t$  participants combine their shares they are able to reconstruct the defining relations for the group  $G$  (see Security 5.1.1) and, since the generating set  $X = \{x_1, x_2, \dots, x_k\}$  is given to each of them, they get the presentation of the group  $G$ , that is,

$$G = \langle x_1, x_2, \dots, x_k \mid r_1 = r_2 = \dots = r_m = 1 \rangle.$$

To reconstruct the secret they solve the word problem for the words  $w_1, w_2, \dots, w_\ell$  in the reconstructed group  $G$ . If  $w_i = 1$  in  $G$ , then  $a_i = 1$  but if  $w_i \neq 1$  in  $G$ , then  $a_i = 0$ .

If less than  $t$  participants combine their shares they get not all of the relations  $r_1, r_2, \dots, r_m$  (see Security 5.1.1). Hence, they obtain a group presentation

$$G_1 = \langle x_1, x_2, \dots, x_k \mid r'_1 = r'_2 = \dots = r'_p = 1 \rangle,$$



with  $r'_i \in \{r_1, r_2, \dots, r_m\}$  for  $1 \leq i \leq p$  and  $p < m$ . Thus, the groups  $G$  and  $G_1$  are not isomorphic and in general  $w_i = 1$  in  $G$  is not equivalent to  $w_i = 1$  in  $G_1$ . Therefore, the participants are not able to reconstruct the correct binary sequence, which is the secret.

**Remark 5.1.4.** The secret (a binary sequence) is not needed for the construction of the shares. It is possible to calculate and to distribute the shares for the participants first and afterwards the secret can be determined and the corresponding sequence of words can be send to the participants. Because of this, the cryptographic protocol can also be used to verify the authenticity of a secret (or a message, respectively). In particular, a secret (or a message, respectively) could contain a predetermined subsequence (a signature) and  $t$  participants may control whether this predetermined sequence is contained in the secret (or the message, respectively) thus validating it. To prevent attacks the place of the signature in a secret (or a message, respectively) should be unknown. We refer to the paper [Pan10] of D. Panagopoulos for description of some more methods for attacks and suggestions for possible group presentations for this cryptographic protocol.

### 5.1.1. Share distribution method given by D. Panagopoulos

We isolate now just the way how D. Panagopoulos distributes elements of a set  $R = \{r_1, r_2, \dots, r_m\}$ , with  $m = \binom{n}{t-1}$  and  $t \leq n$ , between  $n$  participants  $p_i$  in such a way, that

- if  $t$  or more (arbitrary) participants come together and combine their  $t$  subsets  $R_i \subseteq R$  they can reconstruct the set  $R$  and
- if  $t - 1$  or less (arbitrary) participants combine their subsets  $R_i \subseteq R$  they do not get the whole set  $R$ , there is at least one element in  $R$  which is not in the union of  $t - 1$  subsets.

Distribution method:

1. Let  $n, t \in \mathbb{N}$ , with  $t \leq n$ , calculate  $m = \binom{n}{t-1}$ . It is  $R = \{r_1, r_2, \dots, r_m\}$ .
2. Let  $A_1, A_2, \dots, A_m$  be an enumeration of subsets of  $\{1, 2, \dots, n\}$  with  $t-1$  elements. Define  $n$  subsets  $R_1, R_2, \dots, R_n$  of the set  $\{r_1, r_2, \dots, r_m\}$  with the property

$$r_j \in R_i \iff i \notin A_j \quad \text{for } j = 1, 2, \dots, m \text{ and } i = 1, 2, \dots, n.$$

3. The participant  $p_i$  gets the share-set  $R_i$ ,  $1 \leq i \leq n$ .

This distribution method fulfills the desired property, to be used for share distribution to get a  $(n, t)$ -secret sharing scheme, see Security 5.1.1. It is  $|R| = m = \binom{n}{t-1}$  and after Remark 5.1.2 it is known, that  $|R_i| = \binom{n-1}{t-1}$ ,  $1 \leq i \leq n$ . As mentioned in Remark 5.1.3 this is equivalent to the solution of Liu (see Problem 5.0.2 and Remark 5.0.3).

## 5.2. A purely combinatorial $(n, t)$ -secret sharing scheme (Protocol 5)

With the share generation of Section 5.1.1 we get **Protocol 5**, a purely combinatorial  $(n, t)$ -secret sharing scheme, whereby the secret is the sum of the multiplicative inverse of elements in the natural numbers.

We introduce **Protocol 5**: The numbers  $n$  and  $t$  are given, whereby  $n$  is the number of participants and  $t$  is the threshold. It is  $n, t \in \mathbb{N}$  with  $t \leq n$ .

Steps for the dealer:

1. First the dealer calculates the number  $m = \binom{n}{t-1}$ .
2. He chooses  $m$  elements  $a_1, a_2, \dots, a_m \in \mathbb{N}$ . The secret  $S$  is the sum

$$S := \sum_{i=1}^m \frac{1}{a_i} \in \mathbb{Q}^+,$$

with  $\mathbb{Q}^+ := \{x \in \mathbb{Q} | x > 0\}$  the set of all positive rational numbers.

3. From the elements  $a_1, a_2, \dots, a_m \in \mathbb{N}$  he constructs the sets  $R_1, R_2, \dots, R_n$  as follows. Let  $A_1, A_2, \dots, A_m$  be an enumeration of subsets of  $\{1, 2, \dots, n\}$  with  $t-1$  elements. Define  $n$  subsets  $R_1, R_2, \dots, R_n$  of the set  $\{a_1, a_2, \dots, a_m\}$  with the property

$$a_j \in R_i \iff i \notin A_j \quad \text{for } j = 1, 2, \dots, m \text{ and } i = 1, 2, \dots, n.$$

4. The participant  $p_i$  gets the set  $R_i$ ,  $1 \leq i \leq n$ , which is his share.

Steps for the participants:

If  $t$  out of  $n$  participants come together they can reconstruct the secret. They first combine their  $t$  private sets  $R_i$  and get by construction the set  $\{a_1, a_2, \dots, a_m\}$ . The secret is the sum of the inverse elements in the reconstructed set, that is,

$$S = \sum_{i=1}^m \frac{1}{a_i}.$$

The cryptographic protocol is summarized in Table 5.1 (page 115).

**Remark 5.2.1.** It is important in terms of practicability, that the dealer calculates and distributes the shares for the participants long before the secret is needed by the participants. Hence, the dealer has enough time to execute the share distribution method and his computational cost should be of no consequence for the cryptographic protocol. If  $t$  participants reconstruct the secret, they add up only  $m$  elements, which is feasible in linear time.

**Variation 5.2.2.** If the dealer needs a special secret  $\tilde{S} \in \mathbb{Q}$  he gives each participant one more element  $x \in \mathbb{Q}$  in each  $R_i$ , with

$$x := \frac{\tilde{S}}{S}.$$

The participants get  $\tilde{S}$  by multiplying the reconstructed secret  $S$  with  $x$ .

**Remark 5.2.3.** Which information gets an eavesdropper when he knows the number  $x$  given by the Variation 5.2.2? The situation is the following.

- The element  $S$ , which the participants generate, is

$$S = \sum_{i=1}^m \frac{1}{a_i} = \frac{p}{q},$$

with the prime factorization  $p = \prod_{i=1}^{z_1} p_i^{\alpha_i}$  and  $q = \prod_{i=1}^{z_2} q_i^{\beta_i}$ , with  $p_i, q_i \in \mathbb{P}$  and  $p_k \neq p_\ell$ , as well as  $q_k \neq q_\ell$  for  $k \neq \ell$ ,  $\mathbb{P}$  set of all prime numbers,  $\alpha_i, \beta_i \in \mathbb{N}$ . The number  $S$  is

Table 5.1.: Summary of **Protocol 5**: Purely combinatorial  $(n, t)$ -secret sharing scheme

( $n, t$ )-secret sharing scheme	
Dealer	Participants $p_1, p_2, \dots, p_n$
Calculate $m = \binom{n}{t-1}$ . Choose $a_1, a_2, \dots, a_m \in \mathbb{N}$ . Construct sets $R_j \subseteq \{a_1, a_2, \dots, a_m\}$ with share distribution method of Section 5.1.1; it is $ R_j  = \binom{n-1}{t-1}$ for $j = 1, 2, \dots, n$ .  Distribute shares to the participants.	<div style="text-align: center;"> <math>R_1 \longrightarrow p_1</math>  <math>R_2 \longrightarrow p_2</math>  <math>\vdots</math>  <math>R_n \longrightarrow p_n</math> </div> <p><math>t</math> participants combine their shares and thus get the set <math>\{a_1, a_2, \dots, a_m\}</math>. The secret is</p> $S = \sum_{i=1}^m \frac{1}{a_i}.$

a reduced fraction, thus  $p$  and  $q$  are co-prime, that is,  $p_i \neq q_j$  for all  $i$  and  $j$ . Let  $P$  be the set of all prime divisors of  $p$ , that is,  $P = \{p_1, p_2, \dots, p_{z_1}\} \subset \mathbb{P}$ , define analogously  $Q = \{q_1, q_2, \dots, q_{z_2}\} \subset \mathbb{P}$ . It is  $P \cap Q = \emptyset$ .

- The secret  $\tilde{S}$ , which the participants want to know is

$$\tilde{S} = \frac{u}{v},$$

with the prime factorization  $u = \prod_{i=1}^{z_3} u_i^{\gamma_i}$  (or  $u = -\prod_{i=1}^{z_3} u_i^{\gamma_i}$  if  $\tilde{S}$  is a negative rational number) and  $v = \prod_{i=1}^{z_4} v_i^{\delta_i}$ ,  $u_i, v_i \in \mathbb{P}$  and  $u_k \neq u_\ell$  as well as  $v_k \neq v_\ell$  for  $k \neq \ell$ ,  $\gamma_i, \delta_i \in \mathbb{N}$ . The number  $\tilde{S}$  is a reduced fraction, thus  $u$  and  $v$  are co-prime, that is,  $u_i \neq v_j$  for all  $i$  and  $j$ . Let  $U$  be the set of all prime divisors of  $u$ , that is,  $U = \{u_1, u_2, \dots, u_{z_3}\} \subset \mathbb{P}$ , define analogously  $V = \{v_1, v_2, \dots, v_{z_4}\} \subset \mathbb{P}$ . It is  $U \cap V = \emptyset$ .

- Let  $x$  be the additional information for all participants. We assume that  $x$  is public. Let

$$\tilde{x} = \frac{uq}{vp} = \frac{r}{t}$$

be uncanceled and  $x$  the canceled fraction of  $\tilde{x}$ . It is  $r = uq$  and  $t = vp$ , that means  $\frac{r}{t}$  is an uncanceled fraction. The numerator of  $\tilde{x}$  has the prime factorization  $r = \prod_{i=1}^{z_5} r_i^{\eta_i}$  (or  $r = -\prod_{i=1}^{z_5} r_i^{\eta_i}$  if  $\tilde{S}$  and hence  $\tilde{x}$  is a negative rational number),  $r_i \in (U \cup Q)$ ,  $\eta_i \in \mathbb{N}$ , and the denominator has the prime factorization  $t = \prod_{i=1}^{z_6} t_i^{\epsilon_i}$ ,  $t_i \in (V \cup P)$ ,  $\epsilon_i \in \mathbb{N}$ .

It is  $z_5 = |U \cup Q| = z_3 + z_2 - |U \cap Q|$ . If  $r_i \in (U \cap Q)$  and  $r_i = q_{k_1} = u_{k_2}$  for a  $k_1$  with  $1 \leq k_1 \leq z_2$  and for a  $k_2$  with  $1 \leq k_2 \leq z_3$ , then  $\eta_i = \beta_{k_1} + \gamma_{k_2}$ . If  $r_i \notin (U \cap Q)$  then  $r_i \in U$

or  $r_i \in Q$ , if  $r_i \in U$  then  $r_i = u_{k_3}$  for a  $k_3$  with  $1 \leq k_3 \leq z_3$ , and hence  $\eta_i = \gamma_{k_3}$  otherwise if  $r_i \in Q$  then  $r_i = q_{k_4}$  for a  $k_4$  with  $1 \leq k_4 \leq z_2$ , and hence  $\eta_i = \beta_{k_4}$ . Analogously considerations hold for  $t$ .

The goal for an attacker is to get the number

$$\tilde{S} = \frac{u}{v}.$$

There are different situations for the canceled fraction  $x = \frac{\tilde{S}}{S} = \frac{r'}{t'}$  of  $\tilde{x} = \frac{u}{v} \frac{q}{p} = \frac{r}{t}$ . With the prime factorizations  $r' = \prod_{i=1}^{z'_5} r_i^{\eta'_i}$ ,  $r'_i \in (U \cup Q)$ ,  $z'_5 \leq z_5$ ,  $\eta'_i \leq \eta_i$ , and  $t' = \prod_{i=1}^{z'_6} t_i^{\epsilon'_i}$ ,  $t'_i \in (V \cup P)$ ,  $z'_6 \leq z_6$ ,  $\epsilon'_i \leq \epsilon_i$ .

1. If  $u_i \in P$  for some  $1 \leq i \leq z_3$  we get two possibilities:
  - a) It is  $u_i = p_j$ , for a  $j$  with  $1 \leq j \leq z_1$ , and  $\gamma_i \leq \alpha_j$ : now  $u_i \neq r'_k$  for any  $1 \leq k \leq z'_5$ . The prime factor  $u_i$  is missing in the prime factorization of  $r'$ . An attacker knows not which prime factor is missing.
  - b) It is  $u_i = p_j$ , for a  $j$  with  $1 \leq j \leq z_1$ , and  $\gamma_i > \alpha_j$ : The prime factor  $u_i$  is in the prime factorization of  $r'$ , that is,  $u_i = r'_k$ , for a  $k$  with  $1 \leq k \leq z'_5$ , but the exponent for  $u_i$ , that is,  $\gamma_i$  is unknown.
2. If  $u_i \notin P$  for some  $1 \leq i \leq z_3$ : Then  $u$  can be found in the prime factorization of  $r'$ .
3. These considerations can be done analogously for the situations with  $v_i \in Q$  and  $v_i \notin Q$  for  $1 \leq i \leq z_4$ .

If an attacker knows the prime factorization of  $r'$  and  $t'$  he gets some hints for the searched secret  $\frac{u}{v}$ . How useful this is depends on the situations 1. to 3. for  $x$  above.

The variation with a special secret  $\tilde{S}$  can be improved as follows.

**Variation 5.2.4.** Everything as above, with the additional public element  $x = \frac{\tilde{S}}{S}$  (as in Variation 5.2.2) with  $\tilde{S} = \frac{u}{v}$ , but the secret is  $S' = u - v \in \mathbb{Z}$  instead of just  $\tilde{S}$ . The dealer should take care, that for  $x$  the situation of 1. a) from above appears for  $u$  or  $v$ . To get the secret  $S'$  there are different possibilities for  $u$  and  $v$ , such that  $S' = u - v$  and there should be at least one for which arises the situation 1. a) of above for at least one  $u_i \in U$  or  $v_j \in V$ .

The security of this method is a consequence of the following.

**Security 5.2.5.** The security depends on the distribution of the shares and is hence analogous to the Security 5.1.1 of D. Panagopoulos share distribution method. If just  $t - 1$  arbitrary sets (or less) of the sets  $R_1, R_2, \dots, R_n$  are combined, there exist a  $j$ , such that the element  $a_j$  is not included in the union of these sets. If just one element  $a_j$  is absent, the participants do not reconstruct the correct sum  $S$ , and hence cannot compute the correct secret. Each  $a_j$  is in each union of at least  $t$  subsets, thus  $t$  participants get the set  $\{a_1, a_2, \dots, a_m\}$  and are able to reconstruct the secret.

**Remark 5.2.6.** If less than  $t$  participants come together, for example  $1 \leq b < t$ , they generate the set  $\{a'_1, a'_2, \dots, a'_y\}$ , with  $a'_i \in \{a_1, a_2, \dots, a_m\}$  and  $y < m$ . They know that the searched secret  $S$  is greater than the inverse sum over their elements in the set of the union of their shares, that is,  $S > \sum_{i=1}^y \frac{1}{a'_i}$ . Thus, they have more information than an outsider. Indeed they are not able to reconstruct the secret  $S$  if they have not all elements  $a_1, a_2, \dots, a_m$ . Certainly, if we are in the situation where we have a special secret with an additional public element  $x$  or the situation of Variation 5.2.4 then  $b$  participants have no more information than an outsider.

**Example 5.2.7.** We perform the steps for a  $(4, 3)$ -secret sharing scheme. It is  $n = 4$  and  $t = 3$ . The dealer follows the steps:

1. He first calculates  $m = \binom{n}{t-1} = \binom{4}{2} = 6$ .
2. The dealer chooses the numbers  $a_1 := 2, a_2 := 16, a_3 := 3, a_4 := 5, a_5 := 4$  and  $a_6 := 7$ . The secret is

$$S := \sum_{i=1}^m \frac{1}{a_i} = \frac{2501}{1680}.$$

a) The six subsets with size 2 of the set  $\{1, 2, 3, 4\}$  are

$$\begin{aligned} A_1 &= \{1, 2\}, & A_2 &= \{1, 3\}, & A_3 &= \{1, 4\}, \\ A_4 &= \{2, 3\}, & A_5 &= \{2, 4\}, & A_6 &= \{3, 4\}. \end{aligned}$$

With help of the  $A_i, i = 1, 2, \dots, 6$ , the dealer gets the sets  $R_1, R_2, R_3$  and  $R_4$ , which contain elements from  $\{a_1, a_2, \dots, a_6\}$ . He puts the element  $a_j$  for which  $i$  is not contained in the set  $A_j$  for  $i = 1, 2, 3, 4$  and  $j = 1, 2, \dots, 6$ , into the set  $R_i$ :

$$\begin{aligned} 1 \notin A_4, A_5, A_6 &\implies R_1 = \{a_4, a_5, a_6\}, \\ 2 \notin A_2, A_3, A_6 &\implies R_2 = \{a_2, a_3, a_6\}, \\ 3 \notin A_1, A_3, A_5 &\implies R_3 = \{a_1, a_3, a_5\}, \\ 4 \notin A_1, A_2, A_4 &\implies R_4 = \{a_1, a_2, a_4\}. \end{aligned}$$

3. The dealer distributes the set  $R_i$  to the participant  $p_i$ , for  $i = 1, 2, 3, 4$ .

If three of the four participants come together, they can calculate the secret  $S$ . For example the participants  $p_1, p_2$  and  $p_3$  have the set

$$\begin{aligned} \tilde{R} &:= R_1 \cup R_2 \cup R_3 \\ &= \{a_4, a_5, a_6\} \cup \{a_2, a_3, a_6\} \cup \{a_1, a_3, a_5\} \\ &= \{a_1, a_2, a_3, a_4, a_5, a_6\}, \end{aligned}$$

and hence get the secret

$$S = \sum_{i=1}^6 \frac{1}{a_i} = \frac{2501}{1680}, \quad \text{with } a_i \in \tilde{R}.$$

If we are now in the situation, that the dealer needs a special secret, for example  $S' = 12$  we use the Variation 5.2.4. The public element  $x$  is of the unreduced form

$$\tilde{x} = \frac{u \cdot 1680}{v \cdot 2501} = \frac{u \cdot 2^4 \cdot 3 \cdot 5 \cdot 7}{v \cdot 41 \cdot 61}$$

with the notations in Remark 5.2.3. The dealer uses for  $u$  the number  $u = 41 \cdot 7 = 287$ , then  $v = 287 - 12 = 275 = 5^2 \cdot 11$  because the secret is  $12 = S' = u - v$ . Now, it is

$$x = \frac{2^4 \cdot 3 \cdot 7^2}{5 \cdot 11 \cdot 61} = \frac{2352}{3355}.$$

After the reconstruction of  $S = \frac{2501}{1680}$  the participants calculate  $x \cdot S = \frac{287}{275}$  and get the secret  $S' = 287 - 275 = 12$ .

For the developed  $(n, t)$ -secret sharing scheme, with the set  $\{a_1, a_2, \dots, a_m\}$ ,  $m = \binom{n}{t-1}$  and  $a_i \in \mathbb{N}$ , the secret is

$$S = \sum_{i=1}^m \frac{1}{a_i}.$$

We could also use some other sums to define a secret. Some examples are:

- Let  $a_i \in \mathbb{Q}$ ,  $1 \leq i \leq m$ . The secret is

$$S = \sum_{i=1}^m a_i.$$

- Let  $a_i \in \{1, 2, \dots, \alpha - 1\}$ ,  $1 \leq i \leq m$ , for a qualified  $\alpha \in \mathbb{N}$ . The secret is

$$S = \sum_{i=1}^m a_i \pmod{\alpha}.$$

This case appears in [BL90], if J. Benaloh and J. Leichter use a minimal CNF form to describe an access structure of a  $(n, t)$ -secret sharing scheme (see Section 5.3.2 for a detailed explanation).

- Let  $a_i \in \mathbb{K}$ ,  $1 \leq i \leq m$  and  $\mathbb{K}$  a field (for example  $\mathbb{K} = \mathbb{Q}$ ) or a big finite field. The secret is

$$S = a_1^2 + a_2^2 + \dots + a_m^2 - a_1 a_2 \dots a_m.$$

This secret is based on the Hurwitz equation, which is

$$x_1^2 + x_2^2 + \dots + x_m^2 = dx_1 x_2 \dots x_m - k,$$

with  $m \geq 3$ ,  $d \in \mathbb{N}$  and  $k \in \mathbb{N} \cup \{0\}$ . This variation of a  $(n, t)$ -secret sharing scheme is published in [FKIMR15].

### 5.3. Access structures for generalized secret sharing schemes

M. Ito, A. Saito and T. Nishizeki in [ISN87] were the first (see [MvOV97]), who studied generalized secret sharing schemes and the idea of access structures. They extend  $(n, t)$ -secret sharing schemes to generalized secret sharing schemes. There is also a second paper [ISN93] from them, which is an extended version of the Paper [ISN87].

They provide in their paper [ISN87] a methodology to design a secret sharing scheme realizing any given monotone access structure. Their secret sharing scheme is a so called multiple assignment scheme, which means, that the dealer distributes to each participant several shares of a  $(n, t)$ -secret sharing scheme. They showed that it is possible to realize any given monotone access structure with the help of a  $(n, t)$ -secret sharing scheme using multiple assignment schemes.

In [BL90] J. Benaloh and J. Leichter describe also a method to construct secret sharing schemes for any given monotone access structure. The construction of M. Ito, A. Saito and T. Nishizeki is generalized by the method of J. Benaloh and J. Leichter, which uses the correspondence between access structures and monotone boolean formulae (see Appendix A.1 for background

information about boolean formulae (or also just called formulae)).

Firstly, in Section 5.3.1, we take a closer look at the construction of M. Ito, A. Saito and T. Nishizeki and we show that in the proof of Theorem 1 in [ISN87] the distribution of shares, which D. Panagopoulos uses and we use for the combinatorial  $(n, t)$ -secret sharing scheme (**Protocol 5**) in Section 5.1.1, is contained as a special case.

Secondly, in Section 5.3.2, we shortly summarize how J. Benaloh and J. Leichter realize any given access structure for a secret sharing scheme with the help of boolean formulae. We then show that the access structure of a  $(n, t)$ -secret sharing scheme of M. Ito, A. Saito and T. Nishizeki corresponds to the case of minimal CNF form (see Definition A.1.9) in the variation of J. Benaloh and J. Leichter. Therefore, we will see that **Protocol 5** is a very similar scheme to the one which J. Benaloh and J. Leichter obtain for  $(n, t)$ -secret sharing schemes using minimal CNF form, this is summarized in Table 5.3 (page 129).

We start with some notations. Let  $M$  be a set, we denote by  $\mathcal{P}(M)$  the power set of  $M$  and by  $|M|$  the cardinality of  $M$ .

**Definition 5.3.1.** Let  $S$  be a secret and  $P = \{p_1, p_2, \dots, p_m\}$ , with  $m \in \mathbb{N}$ , the set of all participants for a secret sharing scheme.

The family  $\{P_j \subseteq P \mid \text{the participants in } P_j \text{ can reconstruct the secret } S\}$  is called the **access structure** of the secret sharing scheme and its elements  $P_j$  are called **qualified subsets**.

Similar to the definition of  $(n, t)$ -secret sharing schemes (see Definition 5.0.1), we get the following definition for (generalized) secret sharing schemes.

**Definition 5.3.2.** Let  $P = \{p_1, p_2, \dots, p_m\}$  be a set of  $m$  participants (or trustees). A **generalized secret sharing scheme** (or just secret sharing scheme) is a method to split a secret  $S$  into  $m$  shares  $s_1, s_2, \dots, s_m$  and distribute each share  $s_i$  to one participant  $p_i$ ,  $1 \leq i \leq m$ , in such a way that

1. if  $P_j = \{p_{j_1}, p_{j_2}, \dots, p_{j_i}\} \subseteq P$  is a qualified subset of participants, then the secret  $S$  can be reconstructed with the help of their shares  $s_{j_1}, s_{j_2}, \dots, s_{j_i}$ ;
2. if  $P_j = \{p_{j_1}, p_{j_2}, \dots, p_{j_i}\} \subseteq P$  is not a qualified subset of participants, then the secret  $S$  cannot be reconstructed with the help of their shares  $s_{j_1}, s_{j_2}, \dots, s_{j_i}$ ;

We next give examples for generalized secret sharing schemes.

**Example 5.3.3.** 1. For a  $(n, t)$ -secret sharing scheme with  $P = \{p_1, p_2, \dots, p_n\}$  and threshold  $t$ , we get the access structure  $\mathcal{A}_{n,t} := \{P_j \subseteq P \mid |P_j| \geq t\}$ . Thus, we understand a  $(n, t)$ -secret sharing scheme as a special case of a generalized secret sharing scheme.

2. An example for an access structure, which belongs not to a  $(n, t)$ -secret sharing scheme and is a generalized secret sharing scheme, is the following. Assume in a company are two directors  $D_1$  and  $D_2$  and three vice-directors  $V_1, V_2$  and  $V_3$ , hence  $P = \{D_1, D_2, V_1, V_2, V_3\}$  is the set of participants. Further, a secret  $S$  can be reconstructed if two directors or three vice-directors or one director and two vice-directors of the company cooperate. Thus, the access structure is

$$\begin{aligned} \mathcal{A} = \{ & \{D_1, D_2\}, \{D_1, V_1, V_2\}, \{D_1, V_1, V_3\}, \{D_1, V_2, V_3\}, \{D_2, V_1, V_2\}, \{D_2, V_1, V_3\}, \\ & \{D_2, V_2, V_3\}, \{V_1, V_2, V_3\}, \{D_1, V_1, V_2, V_3\}, \{D_2, V_1, V_2, V_3\}, \\ & \{D_1, D_2, V_1, V_2, V_3\}, \{D_1, D_2, V_1\}, \{D_1, D_2, V_2\}, \\ & \{D_1, D_2, V_3\}, \{D_1, D_2, V_1, V_2\}, \{D_1, D_2, V_1, V_3\}, \{D_1, D_2, V_2, V_3\} \}. \end{aligned}$$

**Proposition 5.3.4.** [ISN87, Proposition 1]

If  $\mathcal{A} \subseteq \mathcal{P}(P)$  is an access structure of a secret sharing scheme, then  $\mathcal{A}$  satisfies

$$A \in \mathcal{A} \wedge A \subseteq A' \subseteq P \implies A' \in \mathcal{A}. \quad (5.1)$$

If  $\mathcal{A} \subseteq \mathcal{P}(P)$  fulfills (5.1) we call it **monotone** (see also [BL90]). Secret sharing schemes with a monotone access structure are also called monotone. All  $(n, t)$ -secret sharing schemes are monotone.

### 5.3.1. Generalized secret sharing schemes by M. Ito, A. Saito and T. Nishizeki

To realize any access structure  $\mathcal{A}$ , which satisfies (5.1) M. Ito, A. Saito and T. Nishizeki use a multiple assignment scheme. In [ISN87] they explain a multiple assignment scheme using A. Shamir's secret sharing scheme, but it is possible to generate multiple assignment schemes with any  $(n, t)$ -secret sharing scheme, which is able to generate  $m \geq n$  shares (this conforms the property (2) which A. Shamir claims for  $(n, t)$ -secret sharing schemes, see Section 5.4). We explain multiple assignment schemes in general.

#### Multiple assignment schemes

Let  $S$  be a secret and  $P = \{p_1, p_2, \dots, p_n\}$  is the set of persons, who are involved in the secret sharing scheme.

1. Choose a  $(n, t)$ -secret sharing scheme, which is able to generate  $m \geq n$  shares.
2. Generate the set  $M = \{s_1, s_2, \dots, s_m\}$ , which is the set of  $m$  shares with the property that arbitrary  $t$  shares of  $M$  can reconstruct the secret  $S$ .
3. Choose  $S_i \subseteq M$ ,  $1 \leq i \leq n$ , and distribute  $S_i$  to the participant  $p_i$ , for  $1 \leq i \leq n$ .

The distribution in step 3. can be considered as a function

$$\begin{aligned} g : P &\rightarrow \mathcal{P}(M) \\ p_i &\mapsto S_i. \end{aligned}$$

Now, the multiple assignment scheme consist of the following access structure

$$\mathcal{A} = \left\{ Q \subseteq P \mid \left| \bigcup_{p_i \in Q} g(p_i) \right| \geq t \right\}.$$

If each  $g(p_i)$  consists of just one element, we are in the situation of a  $(n, t)$ -secret sharing scheme with  $n = |P|$ . Thus, a multiple assignment scheme is a generalization of a  $(n, t)$ -secret sharing scheme.

**Definition 5.3.5.** For a finite set  $P$ , a finite set  $M$ , a function  $g : P \rightarrow \mathcal{P}(M)$  and a natural number  $t$  we define  $\mathcal{A}(g, t)$  as

$$\mathcal{A}(g, t) := \left\{ Q \subseteq P \mid \left| \bigcup_{p_i \in Q} g(p_i) \right| \geq t \right\}.$$

**Remark 5.3.6.** If  $M$  is a set of shares for a  $(n, t)$ -secret sharing scheme with  $|M| = n$ , then the monotone access structure  $\mathcal{A}(g, t)$  is exactly a monotone access structure of the multiple assignment scheme defined by  $M$  and the assignment  $g : P \rightarrow \mathcal{P}(M)$ .



In the proof of the next theorem M. Ito, A. Saito and T. Nishizeki explain how each monotone access structure can be realized as an access structure for a multiple assignment scheme.

**Theorem 5.3.7.** [ISN87, Theorem 1]

Let  $P$  be a set of participants. For any  $\mathcal{A} \subseteq \mathcal{P}(P)$  satisfying (5.1), there exists a set  $M$ , a function  $g : P \rightarrow \mathcal{P}(M)$  and a natural number  $t$ , such that  $\mathcal{A}(g, t) = \mathcal{A}$ .

They use in the proof of Theorem 5.3.7 a construction to realize an access structure  $\mathcal{A} = \mathcal{A}(g, t)$  from information of the unqualified subsets, as follows.

**Construction in the proof of Theorem 5.3.7:**

Let  $\mathcal{A}$  be a monotone access structure (that means it satisfies (5.1)). Let  $\mathcal{B} = \mathcal{P}(M) \setminus \mathcal{A}$ . The elements in the family  $\mathcal{B}$  are called **unqualified subsets** and satisfies by (5.1) the property

$$B \in \mathcal{B} \wedge B' \subseteq B \implies B' \in \mathcal{B}. \quad (5.2)$$

**Remark 5.3.8.** In general a **perfect** secret sharing scheme is a scheme in which any unqualified subset of  $P$  cannot get any information about the secret  $S$ .

The family of maximal sets in  $\mathcal{B}$  is denoted by  $\partial^+\mathcal{B}$ , it is

$$\partial^+\mathcal{B} := \{B \in \mathcal{B} \mid B \not\subseteq B' \text{ for all } B' \in \mathcal{B} \setminus \{B\}\}.$$

A set  $M$  is constructed, such that each element of  $M$  corresponds one-to-one to a maximal set of  $\mathcal{B}$ . Thus,  $M$  is defined by

$$M := \{s_B \mid B \in \partial^+\mathcal{B}\},$$

where  $s_B \neq s_{B'}$ , if  $B, B' \in \partial^+\mathcal{B}$  and  $B \neq B'$ . The function  $g$  is defined by  $g : P \rightarrow \mathcal{P}(M)$  with

$$g(p) = \{s_B \mid B \in \partial^+\mathcal{B}, p \notin B\}$$

and it is shown in the next proof that now  $\mathcal{A}(g, t) = \mathcal{A}$  if  $t = |\partial^+\mathcal{B}| = |M|$ .

*Proof.* [ISN87, Proof of Theorem 1]:

To show  $\mathcal{A}(g, t) = \mathcal{A}$  if  $t = |\partial^+\mathcal{B}| = |M|$  we show first  $\mathcal{A} \subseteq \mathcal{A}(g, t)$  if  $t = |\partial^+\mathcal{B}| = |M|$ : Assume the contrary, that means,  $\mathcal{A} \not\subseteq \mathcal{A}(g, t)$  and  $t = |\partial^+\mathcal{B}| = |M|$ , hence there exists a  $Q \in \mathcal{A}$  but  $Q \notin \mathcal{A}(g, t)$ . Since  $t = |M|$  it is

$$\bigcup_{p \in Q} g(p) \neq M,$$

therefore it exists an element  $B \in \partial^+\mathcal{B}$ , such that

$$s_B \in M \setminus \bigcup_{p \in Q} g(p).$$

Hence, for each  $p \in Q$  it is  $s_B \notin g(p)$  and after definition of the function  $g$  it is  $p \in B$  and thus  $Q \subseteq B$ . After (5.2) we get  $Q \in \mathcal{B}$ , because  $B \in \mathcal{B}$  and  $Q \subseteq B$ . Thus,  $Q \in \mathcal{B}$  and  $Q \in \mathcal{A}$  which contradicts the definition of  $\mathcal{B} := \mathcal{P}(M) \setminus \mathcal{A}$ . Hence, it is  $\mathcal{A} \subseteq \mathcal{A}(g, t)$  if  $t = |\partial^+\mathcal{B}| = |M|$ .

Next, we show  $\mathcal{A}(g, t) \subseteq \mathcal{A}$  if  $t = |\partial^+\mathcal{B}| = |M|$ : Assume the contrary, that means,  $\mathcal{A}(g, t) \not\subseteq \mathcal{A}$  and  $t = |\partial^+\mathcal{B}| = |M|$ . Thus, there exists a  $Q \in \mathcal{A}(g, t)$  but  $Q \notin \mathcal{A}$ . After the definition of  $\mathcal{B}$ , it is  $Q \in \mathcal{B}$  if  $Q \notin \mathcal{A}$ . Therefore,  $Q \in \mathcal{B}$  and so  $Q \subseteq B$  for some  $B \in \partial^+\mathcal{B}$ . For all  $p \in Q \subseteq B$  it is

$$s_B \notin \bigcup_{p \in Q} g(p)$$

by definition of  $g$  and hence  $Q \notin \mathcal{A}(g, t)$ , which contradicts the assumption  $Q \in \mathcal{A}(g, t)$ . Hence, it is  $\mathcal{A}(g, t) \subseteq \mathcal{A}$  if  $t = |\partial^+ \mathcal{B}| = |M|$ .

Altogether it is shown that  $\mathcal{A}(g, t) = \mathcal{A}$  if  $t = |\partial^+ \mathcal{B}| = |M|$ .  $\square$

**Remark 5.3.9.** By Theorem 5.3.7, a family  $\mathcal{A} \subseteq \mathcal{P}(P)$  is called an **access structure (of some multiple assignment scheme)** if  $\mathcal{A}$  satisfies (5.1).

**Remark 5.3.10.** If an arbitrary monotone access structure  $\mathcal{A}$  is given we are able to realize this access structure with the help of the construction of Theorem 5.3.7. If we define  $t = |\partial^+ \mathcal{B}| = |M| =: m'$  we get  $\mathcal{A}(g, m') = \mathcal{A}$ . If  $M$  is the set of shares for a  $(m', m')$ -secret sharing scheme then  $\mathcal{A}(g, m')$  is exactly an access structure of the multiple assignment scheme defined by  $M$  and  $g$ , whereby  $g$  is defined as in the construction for Theorem 5.3.7 and we get  $\mathcal{A}(g, m') = \mathcal{A}$ . Thus, also hierarchical secret sharing schemes (which are monotone) can be realized. This was claimed by A. Shamir in property (4), see Section 5.4. Hierarchical secret sharing schemes are also known as asymmetric secret sharing schemes.

Now, we show how the share distribution method given by D. Panagopoulos is given by M. Ito, A. Saito and T. Nishizeki. Let  $P = \{p_1, p_2, \dots, p_n\}$  be the set of participants. If we realize the access structure  $\mathcal{A}_{n,t} = \{P' \subseteq P \mid |P'| \geq t\}$  with the help of a multiple assignment scheme with a  $(m', m')$ -secret sharing scheme, as described above for Theorem 5.3.7, we get the share distribution method given by D. Panagopoulos (see Section 5.1.1).

It is shown (proof of Theorem 5.3.7) that  $\mathcal{A}(g, m') = \mathcal{A}_{n,t}$  if  $m' = |\partial^+ \mathcal{B}| = |M|$  and if  $M$  is a set of shares for a  $(|M|, m')$ -secret sharing scheme, then  $\mathcal{A}(g, m')$  is exactly an access structure of the multiple assignment scheme defined by  $M$  and  $g$ .

We first need the family  $\mathcal{B}$  of all unqualified subsets for the access structure  $\mathcal{A}_{n,t}$ . After definition it is  $\mathcal{B} = \mathcal{P}(P) \setminus \mathcal{A}_{n,t}$  and hence

$$\mathcal{B} = \{P' \subseteq P \mid |P'| \leq t - 1\}.$$

The family of maximal sets in  $\mathcal{B}$  is, after definition,

$$\partial^+ \mathcal{B} = \{P' \subseteq P \mid |P'| = t - 1\}.$$

It is  $|P| = n$  and there are  $\binom{n}{t-1}$  subsets of  $P$  with  $t - 1$  elements. Hence,  $|\partial^+ \mathcal{B}| = \binom{n}{t-1}$ , we get  $m' = \binom{n}{t-1}$

Because  $m' = |M| = |\partial^+ \mathcal{B}|$ , it is  $\partial^+ \mathcal{B} = \{P'_1, P'_2, \dots, P'_{m'}\}$ ,  $M = \{s_1, s_2, \dots, s_{m'}\}$  and we get the one-to-one assignment  $P'_i \xrightarrow{1:1} s_i$ ,  $1 \leq i \leq m'$  with  $s_j \neq s_i$ , if  $P'_i, P'_j \in \partial^+ \mathcal{B}$  but  $P'_j \neq P'_i$ .

The assignment  $g$  is now  $g : P \rightarrow \mathcal{P}(M)$  with  $g(p) = \{s_j \mid P'_j \in \partial^+ \mathcal{B}, p \notin P'_j\}$  in other words, we give participant  $p_i$  element  $s_j$  if and only if  $p_i$  is not an element in  $P'_j$ , for all  $1 \leq j \leq m'$  and  $1 \leq i \leq n$ .

This is exactly what D. Panagopoulos describes in [Pan10] for the way how he constructs the shares  $R_i$ , which are subsets of  $R = \{r_1, r_2, \dots, r_m\}$ , with  $m = \binom{n}{t-1}$ , for his  $(n, t)$ -secret sharing scheme, and distributes them to the participants  $p_i$ ,  $1 \leq i \leq n$ . We shortly summarize both methods in Table 5.2 (page 123).

Table 5.2.: D. Panagopoulos' share distribution method and M. Ito, A. Saito and T. Nishizeki construction for a  $(n, t)$ -secret sharing scheme using multiple assignment scheme

<p><b>M. Ito, A. Saito and T. Nishizeki:</b> construction to realize an access structure <math>\mathcal{A}_{n,t}</math> (for a <math>(n, t)</math>-secret sharing scheme) with the help of a multiple assignment scheme</p>	<p><b>D. Panagopoulos:</b> construction of shares for his <math>(n, t)</math>-secret sharing scheme</p>
<p><math>P = \{p_1, p_2, \dots, p_n\}</math> set of participants</p>	<p><math>P = \{p_1, p_2, \dots, p_n\}</math> set of participants</p>
<p>They calculate <math>m'</math> with <math>\partial^+ \mathcal{B} = \{P' \subseteq P \mid  P'  = t - 1\}</math>, hence it is <math>m' :=  M  =  \partial^+ \mathcal{B}  = \binom{n}{t-1}</math>.</p> $m' = m$	<p>He defines <math>m := \binom{n}{t-1}</math>.</p>
<p><math>P'_i \in \partial^+ \mathcal{B}</math>, <math>P'_i</math> subsets of <math>P</math> of size <math>t - 1</math>, <math>1 \leq i \leq m'</math></p> <p><math>p_j \in P</math></p> $P'_i \xleftrightarrow{1:1} A_i$ $p_j \xleftrightarrow{1:1} j$	<p><math>A_i</math> subsets of <math>\{1, 2, \dots, n\}</math> with <math>t - 1</math> elements, <math>1 \leq i \leq m</math></p> <p><math>j \in \{1, 2, \dots, n\}</math></p>
<p><math>M = \{s_1, s_2, \dots, s_{m'}\}</math></p> $s_i \xleftrightarrow{1:1} r_i$	<p><math>R = \{r_1, r_2, \dots, r_m\}</math></p>
<p>Assignment to distribute the elements of <math>M</math> to the participants:  <math>g : P \rightarrow \mathcal{P}(M)</math> with  <math>g(p) = \{s_j \mid P'_j \in \partial^+ \mathcal{B}, p \notin P'_j\}</math>,          this is equivalent to</p> $s_j \in g(p_i) \iff p_i \notin P'_j,$ <p><math>1 \leq j \leq m'</math> and <math>1 \leq i \leq n</math>.          The participant <math>p_i</math> gets the set <math>g(p_i)</math>.</p> $g(p_i) \xleftrightarrow{1:1} R_i$	<p>Construct set <math>R_i</math> for the participant <math>p_i</math>, <math>i = 1, 2, \dots, n</math>, with</p> $r_j \in R_i \iff i \notin A_j,$ <p><math>1 \leq j \leq m</math> and <math>1 \leq i \leq n</math>.          The participant <math>p_i</math> gets the set <math>R_i</math>.</p>

**Remark 5.3.11.** In a  $(n, t)$ -secret sharing scheme, which is realized with a multiple assignment scheme, which uses for the shares a  $(m, m)$ -secret sharing scheme (with  $m = \binom{n}{t-1}$  see above), each participant gets  $\binom{n-1}{t-1}$  shares of the used  $(m, m)$ -secret sharing scheme, that is,  $|g(p_i)| = \binom{n-1}{t-1}$  for all  $i = 1, 2, \dots, n$ . Because of the assignment  $g$  each participant gets a share for each subset of size  $t-1$  in which he is not a member. If  $|P| = n$ , then there are exactly  $\binom{n-1}{t-1}$  subsets of  $P$  in which the participant  $p_i \in P$  is not a member.

With the multiple assignment scheme  $(n, t)$ -secret sharing schemes can be used to realize asymmetric (or hierarchical, see for instance [Sha79]) secret sharing schemes. If we use the construction of Theorem 5.3.7, we can realize any given monotone access structure, but this construction is not the only way to realize a given monotone access structure, it is possible that there exists also other constructions, see Example 5.3.12.

**Example 5.3.12.** Assume in a company are two directors  $D_1$  and  $D_2$  and three vice-directors  $V_1, V_2$  and  $V_3$ , hence  $P = \{D_1, D_2, V_1, V_2, V_3\}$  is the set of participants. Further, a secret  $S$  can be reconstructed if two directors or three vice-directors or one director and two vice-directors of the company cooperate. Thus, the monotone access structure is

$$\begin{aligned} \mathcal{A} = \{ & \{D_1, D_2\}, \{D_1, V_1, V_2\}, \{D_1, V_1, V_3\}, \{D_1, V_2, V_3\}, \{D_2, V_1, V_2\}, \{D_2, V_1, V_3\}, \{D_2, V_2, V_3\}, \\ & \{V_1, V_2, V_3\}, \{D_1, V_1, V_2, V_3\}, \{D_2, V_1, V_2, V_3\}, \{D_1, D_2, V_1, V_2, V_3\}, \{D_1, D_2, V_1\}, \\ & \{D_1, D_2, V_2\}, \{D_1, D_2, V_3\}, \{D_1, D_2, V_1, V_2\}, \{D_1, D_2, V_1, V_3\}, \{D_1, D_2, V_2, V_3\} \}. \end{aligned}$$

Therefore, the set of unqualified subsets is  $\mathcal{B} = \mathcal{P}(P) \setminus \mathcal{A}$ , with

$$\begin{aligned} \mathcal{B} = \{ & \{D_1\}, \{D_2\}, \{D_1, V_1\}, \{D_1, V_2\}, \{D_1, V_3\}, \{D_2, V_1\}, \{D_2, V_2\}, \{D_2, V_3\}, \\ & \{V_1\}, \{V_2\}, \{V_3\}, \{V_1, V_2\}, \{V_1, V_3\}, \{V_2, V_3\}, \emptyset \}. \end{aligned}$$

Now, we give two possibilities to realize the access structure  $\mathcal{A}$ . Firstly, we use shares of a  $(n, t)$ -secret sharing scheme. Secondly, we use the construction of Theorem 5.3.7 and hence we need the help of a multiple assignment scheme with a  $(m', m')$ -secret sharing scheme, with  $m' = |\partial^+ \mathcal{B}| = |M|$  and  $M$  the set of the shares for the used  $(m', m')$ -secret sharing scheme.

1. possibility: Use a  $(12, 6)$ -secret sharing scheme. The set of shares is  $M = \{s_1, s_2, \dots, s_{12}\}$ . Each participant gets a subset of the set  $M$  as follows

$$\begin{aligned} D_1 & \rightarrow \{s_1, s_2, s_3\}, & D_2 & \rightarrow \{s_4, s_5, s_6\}, \\ V_1 & \rightarrow \{s_7, s_8\}, & V_2 & \rightarrow \{s_9, s_{10}\} & \text{and} & V_3 & \rightarrow \{s_{11}, s_{12}\}. \end{aligned}$$

How many shares each participant gets depends on the importance of him. Each set of participants in  $\mathcal{A}$  can reconstruct the secret, because they get at least six shares for the  $(12, 6)$ -secret sharing scheme. No set of participants in  $\mathcal{B}$  can reconstruct the secret, because they get at most five different shares from  $M$ .

2. possibility: First we need the set of all maximal unqualified subsets

$$\begin{aligned} \partial^+ \mathcal{B} = \{ & \{D_1, V_1\}, \{D_1, V_2\}, \{D_1, V_3\}, \{D_2, V_1\}, \{D_2, V_2\}, \\ & \{D_2, V_3\}, \{V_1, V_2\}, \{V_1, V_3\}, \{V_2, V_3\} \}. \end{aligned}$$

It is  $|\partial^+ \mathcal{B}| = 9$ , thus  $|M| = 9 = t$ . To get  $\mathcal{A}(g, t)$  with  $\mathcal{A}(g, t) = \mathcal{A}$  we use a  $(9, 9)$ -secret sharing scheme with the share-set  $M = \{s_1, s_2, \dots, s_9\}$  and the following one-to-one

assignment between elements in  $M$  and elements in  $\partial^+\mathcal{B}$

$$\begin{aligned} s_1 &\xleftrightarrow{1:1} \{D_1, V_1\} =: B_1, & s_2 &\xleftrightarrow{1:1} \{D_1, V_2\} =: B_2, & s_3 &\xleftrightarrow{1:1} \{D_1, V_3\} =: B_3, \\ s_4 &\xleftrightarrow{1:1} \{D_2, V_1\} =: B_4, & s_5 &\xleftrightarrow{1:1} \{D_2, V_2\} =: B_5, & s_6 &\xleftrightarrow{1:1} \{D_2, V_3\} =: B_6, \\ s_7 &\xleftrightarrow{1:1} \{V_1, V_2\} =: B_7, & s_8 &\xleftrightarrow{1:1} \{V_1, V_3\} =: B_8, & s_9 &\xleftrightarrow{1:1} \{V_2, V_3\} =: B_9. \end{aligned}$$

With the assignment  $g : P \rightarrow \mathcal{P}(M)$  with  $g(p) = \{s_j \mid B_j \in \partial^+\mathcal{B}, p \notin B_j\}$  it is

$$\begin{aligned} D_1 &\mapsto g(D_1) = \{s_4, s_5, s_6, s_7, s_8, s_9\}, \\ D_2 &\mapsto g(D_2) = \{s_1, s_2, s_3, s_7, s_8, s_9\}, \\ V_1 &\mapsto g(V_1) = \{s_2, s_3, s_5, s_6, s_9\}, \\ V_2 &\mapsto g(V_2) = \{s_1, s_3, s_4, s_6, s_8\}, \\ V_3 &\mapsto g(V_3) = \{s_1, s_2, s_4, s_5, s_7\}. \end{aligned}$$

Each set of participants in  $\mathcal{A}$  can reconstruct the secret  $S$  but no set in  $\mathcal{B}$ , because a  $(9, 9)$ -secret sharing scheme for the multiple assignment scheme and the assignment  $g$  is used to realize the access structure  $\mathcal{A}(g, 9) = \mathcal{A}$ , after Theorem 5.3.7.

**Remark 5.3.13.** A multiple assignment scheme for a given access structure  $\mathcal{A}$  constructed with the construction of the proof of Theorem 5.3.7 uses  $|\partial^+\mathcal{B}| = m'$  shares. The number  $|\partial^+\mathcal{B}| = m'$  could become very large compared with the number of participants for the access structure  $\mathcal{A}$ . In Example 5.3.12, we use an access structure which has 5 participants and it is  $|\partial^+\mathcal{B}| = m' = 9$ . If we use as access structure  $\mathcal{A}_{n,t}$ , which is an access structure for a  $(n, t)$ -secret sharing scheme, we have  $n$  participants and it is  $|\partial^+\mathcal{B}| = \binom{n}{t-1}$ .

In general the theorem of Sperner (Theorem 5.3.15) gives a bound for  $|\partial^+\mathcal{B}|$ , which was also mentioned in [ISN93]. Before we are able to present Sperners Theorem and the subsequently bound, we need the next definition.

**Definition 5.3.14.** Let  $P = \{p_1, p_2, \dots, p_n\}$  be a set of  $n$  elements. A family  $\mathcal{D} \subseteq \mathcal{P}(P)$  is called an **antichain** if

$$D \not\subseteq D' \quad \text{for any } D, D' \in \mathcal{D}.$$

**Theorem 5.3.15.** Sperners Theorem ([Spe82] or [Sac96])

Let  $\mathcal{D}$  be an antichain of a set with  $n$  elements. The number  $m = |\mathcal{D}|$  satisfies the inequality

$$m \leq \binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

Note, for  $r \in \mathbb{R}$  it is  $\lfloor r \rfloor := \max\{x \in \mathbb{Z} \mid x \leq r\}$ .

If  $\mathcal{A}$  is a monotone access structure and  $\mathcal{B} = \mathcal{P}(P) \setminus \mathcal{A}$ , then  $\partial^+\mathcal{B}$  is an antichain and after Sperners Theorem (Theorem 5.3.15) it is

$$|\partial^+\mathcal{B}| \leq \binom{|P|}{\lfloor \frac{|P|}{2} \rfloor}.$$

### 5.3.2. Generalized secret sharing schemes by J. Benaloh and J. Leichter

In [BL90] J. Benaloh and J. Leichter translate the access structure for a monotone secret sharing scheme into a monotone formula. Each variable in this formula is assigned to exactly one

participant in  $P$ , the set of the participants for the desired secret sharing scheme. It is known, that a monotone formula contains only AND and OR operators (Remark A.1.10). Hence, they show how to divide a secret “across” each of these two operators (they make this method more efficient by using also THRESHOLD operators (see [BL90]), but here this is of no interest at the moment).

We now explain how they use the AND and OR operator.

Let  $P = \{p_1, p_2, \dots, p_n\}$  be the set of participants for a generalized secret sharing scheme. Let  $S$  be the secret with  $S \in \{0, 1, 2, \dots, \alpha - 1\}$  for a qualified  $\alpha \in \mathbb{N}$ .

- If a secret should be divide across an AND operator they use the simple sum (see also “unanimous consent control by modular addition” in [MvOV97]). For example the secret should be reconstructible if “ $p_1$  and  $p_2$  and  $\dots$  and  $p_z$ ” ( $z \leq n$ ) come together, the variables for the corresponding formula are assigned to the participants, that means variable  $\tilde{p}_i$  is assigned to participant  $p_i$ . The formula for this access structure is  $\tilde{p}_1 \wedge \tilde{p}_2 \wedge \dots \wedge \tilde{p}_z$ . Choose the shares  $s_i \in \{0, 1, 2, \dots, \alpha - 1\}$  for  $i = 1, 2, \dots, z - 1$  randomly. Calculate

$$s_z = S - \sum_{j=1}^{z-1} s_j \pmod{\alpha}$$

and give each participant  $p_i$  his share  $s_i$ .

- If a secret should be divide across an OR operator they give each set of participant (which is assigned to one literal) the secret  $S$ . For example we assign one literal  $\tilde{p}_i$  to one participant  $p_i$ . If “ $p_1$  or  $p_2$  or  $\dots$  or  $p_q$ ” should be able to reconstruct the secret, then each participant  $p_i$ ,  $1 \leq i \leq q$ , gets the secret. The formula for the access structure is  $\tilde{p}_1 \vee \tilde{p}_2 \vee \dots \vee \tilde{p}_q$ .

Moving a secret across an OR operator corresponds to a  $(n, 1)$ -secret sharing scheme and moving the secret across an AND operator corresponds to a  $(m, m)$ -secret sharing scheme, which are for J. Benaloh and J. Leichter the simple sums. Thus, this can also be seen as a multiple assignment scheme.

We now focus on access structures for  $(n, t)$ -secret sharing schemes. The formula for an access structure  $\mathcal{A}_{n,t}$  can be written in a Disjunctive Normal Form (DNF) (see Definition A.1.7) and also in a logically equivalent Conjunctive Normal Form (CNF) (see Definition A.1.6). In addition a variation is also to use minimal DNF form (see Definition A.1.8) or minimal CNF form (see Definition A.1.9) for formula of an access structures.

**Example 5.3.16.** Let  $P = \{p_1, p_2, p_3\}$  be the set of participants for a  $(3, 2)$ -secret sharing scheme. The access structure is

$$\mathcal{A}_{3,2} = \{P' \subseteq P \mid |P'| \geq 2\} = \{\{p_1, p_2\}, \{p_1, p_3\}, \{p_2, p_3\}, \{p_1, p_2, p_3\}\}.$$

The formula corresponding to this access structure is

$$(\tilde{p}_1 \wedge \tilde{p}_2) \vee (\tilde{p}_1 \wedge \tilde{p}_3) \vee (\tilde{p}_2 \wedge \tilde{p}_3) \vee (\tilde{p}_1 \wedge \tilde{p}_2 \wedge \tilde{p}_3). \quad (5.3)$$

This is in DNF. The secret is first divided “across” the OR operators. After this the secret is shared independently for the AND parts (that is, for each set in the access structure  $\mathcal{A}$ ). Each participant gets one share for each set in the assecc structure in which he is a member. If in this example the secret is  $S = 35$  and  $\alpha = 812$  the dealer does the following for each term in the DNF formula

- 1. term:  $(\tilde{p}_1 \wedge \tilde{p}_2)$ . Divide  $S = 35$  between  $p_1$  and  $p_2$ . Participant  $p_1$  gets the share  $s_{1_1} = 10$  and participant  $p_2$  gets the share  $s_{1_2} = 25$ .
- 2. term:  $(\tilde{p}_1 \wedge \tilde{p}_3)$ . Divide  $S = 35$  between  $p_1$  and  $p_3$ . Participant  $p_1$  gets the share  $s_{2_1} = 30$  and participant  $p_3$  gets the share  $s_{2_3} = 5$ .
- 3. term:  $(\tilde{p}_2 \wedge \tilde{p}_3)$ . Divide  $S = 35$  between  $p_2$  and  $p_3$ . Participant  $p_2$  gets the share  $s_{3_2} = 7$  and participant  $p_3$  gets the share  $s_{3_3} = 28$ .
- 4. term:  $(\tilde{p}_1 \wedge \tilde{p}_2 \wedge \tilde{p}_3)$ . Divide  $S = 35$  between  $p_1, p_2$  and  $p_3$ . Participant  $p_1$  gets the share  $s_{4_1} = 2$ , participant  $p_2$  gets the share  $s_{4_2} = 13$  and participant  $p_3$  gets the share  $s_{4_3} = 20$ .

Altogether participant  $p_1$  holds the multiple share  $S_1 = \{s_{1_1}, s_{2_1}, s_{4_1}\}$ , participant  $p_2$  holds the multiple share  $S_2 = \{s_{1_2}, s_{3_2}, s_{4_2}\}$  and participant  $p_3$  holds the multiple share  $S_3 = \{s_{2_3}, s_{3_3}, s_{4_3}\}$ . If all shares  $s_{i_j}$  with the same  $i$  are combined, then the secret  $S$  is the sum of these elements. The minimal DNF form is here

$$(\tilde{p}_1 \wedge \tilde{p}_2) \vee (\tilde{p}_1 \wedge \tilde{p}_3) \vee (\tilde{p}_2 \wedge \tilde{p}_3),$$

which reduces the shares in the set  $S_i$  for the participants  $p_i$  about one element, precisely  $s_{4_i}$  for participant  $p_i$ , compared with the corresponding shares for the DNF formula (5.3).

It is not clear, that a DNF or a CNF of a formula for an access structure is the most efficient solution to generate the multiple shares for the participants (in the sense that a participant holds as less shares as possible). Even more if J. Benaloh and J. Leichter also add the THRESHOLD operator (for more details see [BL90]).

J. Benaloh and J. Leichter pointed out, that the method of M. Ito, A. Saito and T. Nishizeki corresponds to the case of minimal CNF form.

In the situation of the access structure for a  $(n, t)$ -secret sharing scheme, this could be seen as follows:

The access structure for a  $(n, t)$ -secret sharing scheme with  $P = \{p_1, p_2, \dots, p_n\}$ , the set of participants, is  $\mathcal{A}_{n,t} = \{P' \subseteq P \mid |P'| \geq t\}$ . Hence, the access structure  $\mathcal{A}_{n,t}$  is monotone it is enough to consider the set  $\mathcal{A}_{n,t}^{min} = \{P' \subseteq P \mid |P'| = t\} \subseteq \mathcal{A}_{n,t}$ , because  $\mathcal{A}_{n,t}^{min}$  is the minimal set, which we need to obtain, with the monotone property (5.1), the access structure  $\mathcal{A}_{n,t}$ . Thus, the DNF formula corresponding to  $\mathcal{A}_{n,t}^{min}$  is a minimal DNF form for  $\mathcal{A}_{n,t}$ .

The Algorithm 5.3.17 gives all  $\binom{n}{t}$  elements of  $\mathcal{A}_{n,t}^{min}$  (line [6]). With line [7] we get all terms for a minimal DNF form which corresponds to the access structure  $\mathcal{A}_{n,t}^{min}$  and hence to  $\mathcal{A}_{n,t}$ .

**Algorithm 5.3.17.** Generate terms for minimal DNF form corresponding to  $\mathcal{A}_{n,t}$

```

[1] for  $i_1$  from 1 to  $n - t + 1$  do
[2]     for  $i_2$  from  $i_1 + 1$  to  $n - t + 2$  do
[3]         for  $i_3$  from  $i_2 + 1$  to  $n - t + 3$  do
[4]             . . .
[5]                 for  $i_t$  from  $i_{t-1} + 1$  to  $n$  do
[6]                     print  $\{\{p_{i_1}, p_{i_2}, p_{i_3}, \dots, p_{i_t}\}\}$ ;
[7]                     print  $\{(\tilde{p}_{i_1} \wedge \tilde{p}_{i_2} \wedge \tilde{p}_{i_3} \wedge \dots \wedge \tilde{p}_{i_t})\}$ ;
[8]                 end;
[9]             . . .
[10]        end;
[11]    end;
[12] end;
    
```

The output sets  $\{p_{i_1}, p_{i_2}, p_{i_3}, \dots, p_{i_t}\}$  (and hence also the terms) are ordered with  $i_1 < i_2 < i_3 < \dots < i_t$  and each set has  $t$  elements (each term has  $t$  variables), after construction. Here the DNF formula is monotone, hence only the use of the distributive property

$$a \vee (b \wedge c) = (a \wedge b) \vee (a \wedge c),$$

for  $a, b, c$  a formula, is required to get a monotone CNF formula.

We are interested in a minimal CNF form (Definition A.1.9), therefore we must know how many variables are at least in a clause of CNF formula in this case. One clause with minimal numbers of variables is the one, which gets from each term in the minimal DNF form a variable, which is also included in the clause or if there are just variables the clause does not include yet, then the variable  $\tilde{p}_{i_j}$  is added to the clause with the property, that  $i_j$  is the smallest index of all variables in the term for which  $\tilde{p}_{i_j}$  is not yet included in the clause. Because of the construction of the terms for the minimal DNF form and the consequence that the sets  $\{p_{i_1}, p_{i_2}, p_{i_3}, \dots, p_{i_t}\}$  are ordered with  $i_1 < i_2 < i_3 < \dots < i_t$  it follows, that the clause with the minimum numbers of variables has  $n - t + 1$  variables (this can be seen by Algorithm 5.3.17 line [1]). All combinations of  $n - t + 1$  variables of  $\{\tilde{p}_1, \tilde{p}_2, \tilde{p}_3, \dots, \tilde{p}_n\}$  are possible. Thus, the following Algorithm 5.3.18, gives all  $\binom{n}{n-t+1}$  clauses for a minimal CNF form corresponding to the access structure  $\mathcal{A}_{n,t}$ .

**Algorithm 5.3.18.** Clauses for minimal CNF form corresponding to  $\mathcal{A}_{n,t}$

```

[1] for  $i_1$  from 1 to  $t$  do
[2]   for  $i_2$  from  $i_1 + 1$  to  $t + 1$  do
[3]     for  $i_3$  from  $i_2 + 1$  to  $t + 2$  do
[4]       . . .
[5]         for  $i_{n-t+1}$  from  $i_{n-t} + 1$  to  $n$  do
[6]           print  $\{(\tilde{p}_{i_1} \vee \tilde{p}_{i_2} \vee \tilde{p}_{i_3} \vee \dots \vee \tilde{p}_{i_{n-t+1}})\}$ ;
[7]         end;
[8]       . . .
[9]     end;
[10]   end;
[11] end;
```

The number of clauses in a minimal CNF form gives the number of shares in which the secret is divided by the method of J. Benaloh and J. Leichter. Here it is  $\tilde{m} = \binom{n}{n-t+1}$ , hence they divide the secret into  $\tilde{m}$  shares  $s_1, s_2, s_3, \dots, s_{\tilde{m}}$ . It is

$$S = \sum_{i=1}^{\tilde{m}} s_i \pmod{\alpha},$$

for a qualified  $\alpha \in \mathbb{N}$ . The shares are distributed to the participants. The terms in the minimal CNF form  $\Phi$  are numerated, it is

$$\Phi = \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \dots \wedge \phi_{\tilde{m}},$$

with  $\phi_i, 1 \leq i \leq \tilde{m}$ , a clause generated by the Algorithm 5.3.18 (line [6]). The share  $s_k$  corresponds to the clause  $\phi_k$ . Hence, each participant  $p_j$  gets the share  $s_k$  if  $\tilde{p}_j$  is a variable in the clause  $\phi_k$ . To get the number of shares each participant gets we must know in how many clauses each variable  $\tilde{p}_i$  ( $1 \leq i \leq \tilde{m}$ ) is included. There are  $\tilde{r} := \binom{n-1}{(n-t+1)-1} = \binom{n-1}{n-t}$  many clauses in which  $\tilde{p}_j$  is a variable, hence the participant  $p_j, 1 \leq j \leq \tilde{m}$ , gets a multiple share which is a subset with  $\binom{n-1}{n-t}$  elements of the set  $\{s_1, s_2, \dots, s_{\tilde{m}}\}$ .



This is very similar to **Protocol 5**, the combinatorial  $(n, t)$ -secret sharing scheme (Section 5.2), which was developed from D. Panagopoulos  $(n, t)$ -secret sharing scheme.

The secret sharing scheme from Section 5.2 uses a set of  $m = \binom{n}{t-1}$  elements from which the shares for the participants are constructed. J. Benaloh and J. Leichter need a set with  $\tilde{m} = \binom{n}{n-t+1} = \binom{n}{t-1}$  elements. Each participant in the combinatorial  $(n, t)$ -secret sharing scheme, which uses the share distribution method given D. Panagopoulos (see Section 5.1.1), gets a set of  $r = \binom{n-1}{t-1}$  element as a share (see Remark 5.1.2). In the method of J. Benaloh and J. Leichter each participant gets  $\tilde{r} = \binom{n-1}{n-t} = \binom{n-1}{t-1}$  elements in his set of shares. Hence, in both schemes the number of shares from which the subsets for the participants are generated is the same and also the number of shares which each participant gets in his share-set is the same. The only difference lies in the kind of shares and in the way how the secret is reconstructed. In the combinatorial scheme the secret is the sum over inverse elements of natural numbers (shares are natural numbers) and in the method of J. Benaloh and J. Leichter the secret is the sum of elements modulo  $\alpha$  (for a qualified  $\alpha$  and the shares are elements of  $\{0, 1, 2, \dots, \alpha - 1\}$ ). This is summarized in the Table 5.3 (page 129).

Table 5.3.: Comparison of the values of the combinatorial  $(n, t)$ -secret sharing scheme (**Protocol 5**) and a  $(n, t)$ -secret sharing scheme by J. Benaloh and J. Leichter using minimal CNF form

Values of the schemes	<b>Protocol 5: purely combinatorial <math>(n, t)</math>-secret sharing scheme</b>	<b>J. Benaloh and J. Leichter's method for a <math>(n, t)</math>-secret sharing scheme</b>
Number of shares	$m = \binom{n}{t-1}$	$\tilde{m} = \binom{n}{n-t+1} = \binom{n}{t-1}$ $m = \tilde{m}$
Share-set	$\{a_1, a_2, \dots, a_m\}$ with $a_i \in \mathbb{N}$	$\{s_1, s_2, \dots, s_{\tilde{m}}\}$ with $s_i \in \{0, 1, 2, \dots, \alpha - 1\}$ for a qualified $\alpha \in \mathbb{N}$
Secret $S$	$S = \sum_{i=1}^m \frac{1}{a_i}$ $S \in \mathbb{Q}^+$	$S = \sum_{i=1}^{\tilde{m}} s_i \pmod{\alpha}$ $S \in \{0, 1, 2, \dots, \alpha - 1\}$
Number of shares from the share-set for each participant	$r = \binom{n-1}{t-1}$	$\tilde{r} = \binom{n-1}{n-t} = \binom{n-1}{t-1}$ $r = \tilde{r}$

## 5.4. Comparison with A. Shamir's suggested properties

A. Shamir lists in his paper [Sha79] the following useful properties for his  $(n, t)$ -secret sharing scheme.

- (1) The size of each piece (which are the shares for the participants) does not exceed the size of the original data (which is the secret).
- (2) When  $t$  is kept fixed, pieces can be dynamically added or deleted (for example, when executives join or leave a company) without affecting the other pieces. (A piece is deleted only when a leaving executive makes it completely inaccessible, even to himself.)
- (3) It is easy to change the pieces (the shares for the participants) without changing the original data (which is the secret). All we need is a new polynomial  $g(x)$  with the same free term. A frequent change of this type can greatly enhance security since the pieces exposed by security breaches cannot be accumulated unless all of them are values of the same edition of the polynomial  $g(x)$ .
- (4) By using tuples of polynomial values as pieces, we can get a hierarchical scheme in which the number of pieces needed to determine the secret depends on their importance. For example, if we give the company's president three values of  $g(x)$ , each vice-president two values of  $g(x)$ , and each executive one value of  $g(x)$ , then a  $(n, 3)$ -threshold scheme enables checks to be signed either by any three executives, or by any two executives one of whom is a vice-president, or by the president alone.

In addition we choose the following fifth property.

- (5) It is easy to change the secret without changing the shares of the participants.

This property is not true for Shamir's secret sharing scheme as it is explained above. Since the supporting points (which are the shares for the participants) fix the polynomial and therefore the constant term (which is the secret).

We analyze the CFRZ-scheme, D. Panagopoulos' scheme and the **Protocol 5**, a purely combinatorial  $(n, t)$ -secret sharing schemes, concerning these 5 points; as we also did it in [FMR13] and [CFMRZ16].

**CFRZ-scheme** of Chapter 2 (see also [FMR13] or [CFMRZ16]):

- (1) The secret is a vector in the subspace  $V$  of the real inner product space  $W$ . A share is a basis vector for the subspace  $V$ . Therefore, the size of each share does not exceed the size of the secret.
- (2) If we fix the number  $t$  of shares (we need at least to reconstruct the secret) we can arbitrarily add or delete many shares. The dealer has to pay attention to the construction that every possible combination of  $t$  shares form a basis for the subspace  $V$ .
- (3) We can change the shares without changing the secret. We need only another subspace  $V' \neq V$ , which contains the secret  $w$ . For this new subspace  $V'$  with dimension  $t$  we can calculate new shares, which are a set of vectors where every arbitrary  $t$  of them form a basis for  $V'$ . A new associated vector  $w^*$  can be constructed as explained in Chapter 2 step 5. for the dealer.

- (4) Every  $(n, t)$ -secret sharing scheme can be converted into a hierarchical secret sharing protocol (see Section 5.3 Remark 5.3.10).

Because of Remark 2.0.3 we get the additional property.

- (5) We can change the secret easily. Every vector in the subspace can be used as a new secret  $w_{new} \in V$  (excluded the shares from the participants) and hence we can calculate the associated vector  $w_{new}^*$  as described in Chapter 2 step 5. for the dealer.

**D. Panagopoulos' scheme** of Section 5.1 (see also [FMR13] or [CFMRZ16]):

- (1) The secret is a binary sequence. The shares are sets  $R_j$  of relations. The length from the relations is not defined. In every set  $R_j$  are  $\binom{n-1}{t-1}$  relations from the group. The size of each share can exceed the size of the secret.
- (2) The dealer creates the shares according to instructions (see for instance the summary in Section 5.1.1). Hence, he cannot add or delete shares, because the way he creates them depends on the number  $m$  of relations and the number  $n$  of participants.
- (3) He can change the shares if he changes the group  $G$ . He has to pay attention to the fact that the sent word in the new group is equivalent to 1 if and only if it is equivalent to 1 in the previous group. Then the secret is not changed.
- (4) Every  $(n, t)$ -secret sharing scheme can be converted into a hierarchical secret sharing protocol (see Section 5.3 Remark 5.3.10).

Because of Remark 5.1.4 we get the additional property.

- (5) The secret, which is a binary sequence, can be changed at every time by sending new words to the participants.

**Protocol 5** of Section 5.2 (see also [CFMRZ16]):

Due to the fact, that this cryptographic protocol uses the share distribution method given by D. Panagopoulos it fulfills the same properties of Shamir's list ((1)-(4)) as D. Panagopoulos' scheme does.

- (1) The secret is the sum over  $m$  elements:  $\sum_{i=1}^m \frac{1}{a_i} \in \mathbb{Q}^+$ . The shares are subsets  $R_j$  of  $\{a_1, a_2, \dots, a_m\}$ ,  $a_i \in \mathbb{N}$ , with  $|R_j| = \binom{n-1}{t-1}$ . Therefore, the size of each piece exceeds the size of the secret.
- (2) We use the method of D. Panagopoulos, hence this property is not valid due to the same reasons as for his secret sharing scheme.
- (3) The shares are subsets of the set  $\{a_1, a_2, \dots, a_m\}$ . If we choose a new set  $\{a'_1, a'_2, \dots, a'_m\}$  with the property  $\sum_{j=1}^m \frac{1}{a_j} = \sum_{j=1}^m \frac{1}{a'_j}$  and give each participant subsets of this new set as a share, then the shares can be changed without changing the secret.
- (4) Every  $(n, t)$ -secret sharing scheme can be converted into a hierarchical secret sharing protocol (see Section 5.3 Remark 5.3.10).
- (5) A secret cannot be changed easily without changing the shares, because it is a sum over all elements in the set  $\{a_1, a_2, \dots, a_m\}$  and hence depends on this set.

The comparison is summarized in Table 5.4 (page 132).

Table 5.4.: Summary of the comparison

Shamir's properties	Shamir's scheme	CFRZ-scheme	D. Panagopoulos' scheme	<b>Protocol 5</b>
(1)	✓	✓	–	–
(2)	✓	✓	–	–
(3)	✓	✓	✓	✓
(4)	✓	✓	✓	✓
Additional property (5)	–	✓	✓	–

With the knowledge of Section 5.3.1 we can interpret D. Panagopoulos'  $(n, t)$ -secret sharing scheme and **Protocol 5** as  $(m, m)$ -secret sharing schemes which are converted into  $(n, t)$ -secret sharing schemes by the help of multiple assignment schemes, which is possible if  $m = \binom{n}{t-1}$ . The number of shares each participant gets is  $\binom{n-1}{t-1}$  (see Remark 5.3.11). This corresponds with the combinatorial solution of Lius problem (see Problem 5.0.2 and Remark 5.0.3).

Now, we take a look at the running time for the participants (see also [FMR13] or [CFMRZ16]).

**Shamir's scheme:** The involved polynomial interpolation has a quadratic running time, that means, if we have  $t$  supporting points we get a complexity of  $\mathcal{O}(t^2)$  (see for instance [EMNW11, Section 9.2.1.]).

**The CFRZ-scheme:** In order to orthonormalize  $t$  linear independent vectors in a real inner product space with dimension  $m$  we have a total running time of  $\mathcal{O}(t^2 m)$  (see [FMR13, Section 2.4]).

In the CFRZ-scheme the variable  $m$  depends on the number  $t$ , because  $m > t$  is postulated. The total running time for this scheme is longer than for Shamir's.

**Panagopoulos' scheme:** The word problem in a Coxeter group, for example, is solvable within quadratic running time, due to the fact, that Coxeter groups are automatic (see [BH93]) and automatic groups have a solvable word problem with a quadratic running time (see [LS77]).

**Protocol 5:** For the reconstruction of the shares the participants only add up  $m$  elements. Therefore, for the participants it is just  $\mathcal{O}(m)$ , where  $m = \binom{n}{t-1}$  is already previously calculated by the dealer, and hence  $m$  is fixed for the participants.

**Remark 5.4.1.** In the special case of a  $(t+1, t)$ -secret sharing scheme the running time depends for **Protocol 5** also only on  $t$  like in Sharmir's scheme:

$$\begin{aligned} m &= \binom{n}{t-1} = \binom{t+1}{t-1} = \binom{t+1}{2} \\ &= \frac{(t+1) \cdot t}{2} = \frac{t^2 + t}{2} \\ &< \frac{2t^2}{2} = t^2. \end{aligned}$$

Hence, the running time is also  $\mathcal{O}(t^2)$ , but as shown above the participants only add up  $m$  elements, which is a very easy operation to reconstruct the secret.

As mentioned above in Remark 5.2.1 it is important in terms of practicability, that the dealer calculates and distributes the shares for the participants in **Protocol 5** long before the secret is needed by the participants. Hence, the dealer has enough time to execute the share distribution method and his computational cost should be of no consequence for the cryptographic protocol. Note, that the dealer has to generate  $m = \binom{n}{t-1}$  shares and uses the share distribution method in Section 5.1.1.

The size of the share-set exceeds the size of the secret but the calculation to reconstruct the secret is very easy and fast.



# Chapter 6

## Secret sharing schemes using Nielsen transformations

Now, we introduce first **Protocol 6** and afterwards **Protocol 7**. Both are secret sharing protocols and are based on Nielsen transformations. **Protocol 6** uses in addition a free subgroup of the special linear group  $SL(2, \mathbb{Q})$  and the secret is a sum over traces of matrices in  $SL(2, \mathbb{Q})$ . **Protocol 7** uses in addition Nielsen reduced sets and the secret is a sum which uses the free length of elements in a Nielsen reduced set.

We present both cryptographic protocols as  $(m, m)$ -secret sharing schemes, because it is possible to modify them to any  $(n, t)$ -secret sharing scheme if the share distribution method of Section 5.1.1 is used and  $m$  is determined as  $m = \binom{n}{t-1}$ . Both developed cryptographic protocols are published in the survey article [CFMRZ16] as research in the field of secret sharing schemes. They are also published in [MR15].

### 6.1. Secret sharing scheme based on Nielsen transformations and $SL(2, \mathbb{Q})$ (Protocol 6)

For **Protocol 6** we consider a finitely generated free group  $F$  as an abstract group but also explicit as a subgroup of the special linear group of all  $2 \times 2$  matrices over  $\mathbb{Q}$ , that is,

$$SL(2, \mathbb{Q}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid a, b, c, d \in \mathbb{Q} \text{ and } ad - bc = 1 \right\}.$$

We use the special linear group over the rational numbers because these numbers can be stored and computed more efficiently on a computer than irrational numbers.

We explain **Protocol 6**, while we take a closer look at the steps for the dealer and the steps for the participants.

Steps for the dealer:

For a  $(m, m)$ -secret sharing scheme,  $m \in \mathbb{N}$ , the dealer does the following steps.

1. He chooses an abstract free generating set  $X$  for the free group  $F$  of rank  $m$ , it is

$$F = \langle X \mid \rangle \quad \text{with } X := \{x_1, x_2, \dots, x_m\}.$$

He also needs an explicit free generating set  $M$ , so it is

$$F = \langle M \mid \rangle \quad \text{with } M := \{M_1, M_2, \dots, M_m\}$$

and  $M_i \in \text{SL}(2, \mathbb{Q})$ . Therefore,  $F$  is a subgroup of  $\text{SL}(2, \mathbb{Q})$ , he chooses  $M_i \in \text{SL}(2, \mathbb{Q})$  and takes care that  $F$  is not a subgroup of  $\text{SL}(2, \mathbb{Z})$ , see Security 6.1.2.

2. With the known matrices in the set  $M$  the secret is

$$S := \sum_{j=1}^m \frac{1}{|a_j|} \in \mathbb{Q}^+, \quad \text{with } a_j := \text{tr}(M_j) \in \mathbb{Q},$$

$\text{tr}(M_j)$  is the trace of the matrix  $M_j := \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \text{SL}(2, \mathbb{Q})$ , that is,  $\text{tr}(M_j) := a + d$ .

If the dealer needs a special secret he can act as described in Variation 5.2.2 or Variation 5.2.4.

3. The dealer constructs the shares for the participants in the following way:

He first applies a regular Nielsen transformation simultaneously for both sets  $X$  and  $M$  to get Nielsen equivalent sets  $U$  and  $N$  to  $X$  and  $M$ , respectively (see Figure 6.1).

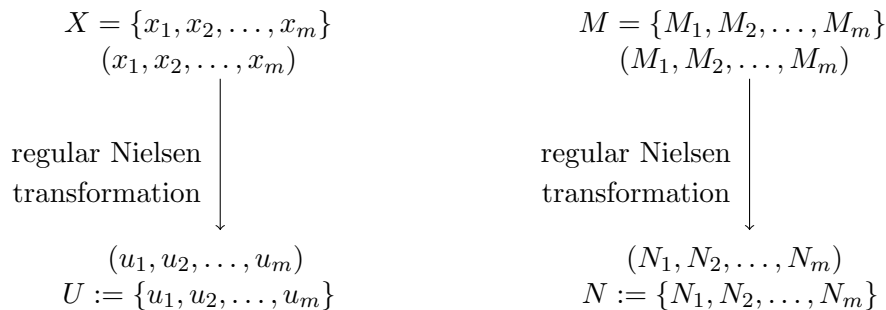


Figure 6.1.: Simultaneously regular Nielsen transformations

The elements  $u_i$  are words in  $X$  and the elements  $N_i$  are words in  $M$ . Hence, we have  $N_i \in \text{SL}(2, \mathbb{Q})$ .

4. Let  $P = \{p_1, p_2, \dots, p_m\}$  be the set of participants for the secret sharing protocol. The dealer distributes to each participant  $p_i$  one abstract share  $u_i$  and one explicit share  $N_i$ . Hence,  $p_i$  gets  $(u_i, N_i)$ ,  $1 \leq i \leq m$ .

If all participants combine their parts they obtain the sets  $U$  and  $N$ .

Steps for the participants:

1. The participants apply regular Nielsen transformations in a Nielsen reducing manner for  $U$  as described in [Ste89], Remark 4.2.17, (or also in [CgRR08] and [LS77]) and step by step simultaneously for  $N$ . By Proposition 4.2.5 and the fact, that the dealer does a regular Nielsen transformation and starts with a basis of  $m$  elements, they get Nielsen reduced sets  $X' = \{x'_1, x'_2, \dots, x'_m\}$  and  $M' = \{M'_1, M'_2, \dots, M'_m\}$ , see Figure 6.2.



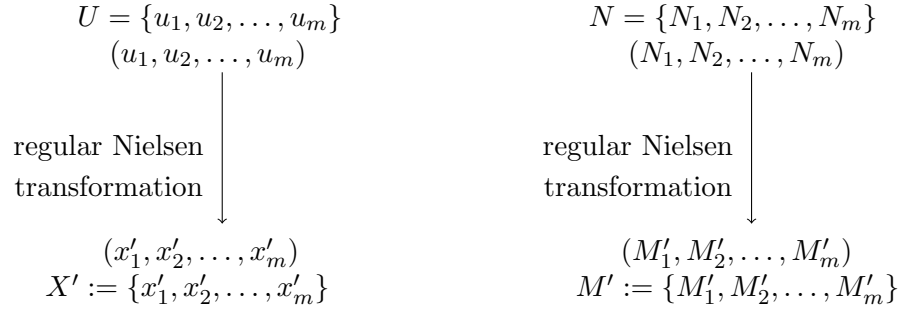


Figure 6.2.: Simultaneously regular Nielsen transformations

Because of Corollary 4.2.9 it is  $X^{\pm 1} = X'^{\pm 1}$  and  $M^{\pm 1} = M'^{\pm 1}$ , respectively. Hence,  $(x'_1, x'_2, \dots, x'_m)$  differs to  $(x_1, x_2, \dots, x_m)$  just in the position order and inverses. That means the set  $X'$  is the set  $X$  up to inverses. The same is true for  $M'$  and  $M$ . Thus, it is  $X' = \{x_1^{\epsilon_1}, x_2^{\epsilon_2}, \dots, x_m^{\epsilon_m}\}$  and  $M' = \{M_1^{\delta_1}, M_2^{\delta_2}, \dots, M_m^{\delta_m}\}$  with  $\epsilon_i, \delta_i \in \{1, -1\}$ .

2. With the knowledge of the set  $M'$  it is easy to reconstruct the secret

$$S = \sum_{j=1}^m \frac{1}{|a_j|} \in \mathbb{Q}^+, \quad \text{with } \text{tr}(M_j) = a_j \in \mathbb{Q}.$$

Recall that here  $\text{tr}(M_i^{\delta_i}) = \text{tr}(M_i)$ ,  $\delta_i \in \{1, -1\}$ , for  $i = 1, \dots, m$ .

If the participants get just matrices of the set  $M$ , from which the secret can be reconstructed, and they know that the secret is a special sum over the traces of the matrices in  $M$ , then they know that the searched secret is bigger than the special sum over their known matrices. Hence, they know more than an outsider and the secret sharing scheme is not perfect in the information-theoretic sense. Therefore, the Nielsen transformation is needed.

**Remark 6.1.1.** If a  $(n, t)$ -secret sharing scheme is desired the dealer chooses  $m = \binom{n}{t-1}$  and uses the method described in Section 5.1.1 for distribution of  $U$  and  $N$  and gives the two share-sets  $R_i$  and  $S_j$  to participant  $p_k$  with  $R_i \subseteq U$  and  $S_j \subseteq N$ ,  $1 \leq k \leq n$ . If  $t$  or more participants combine their shares, they get the sets  $U$  and  $N$ . With these sets they are able to reconstruct the secret as explained above in the steps for the participants. Less than  $t$  participants can neither get the whole set  $U$ , which is Nielsen equivalent to  $X'$ , nor the whole set  $N$ , which is Nielsen equivalent to  $M'$ .

The cryptographic protocol as a  $(n, t)$ -secret sharing scheme is summarized in Table 6.1 (page 138).

Table 6.1.: Summary of **Protocol 6**: Secret sharing scheme using Nielsen transformations and  $\text{SL}(2, \mathbb{Q})$

( $n, t$ )-secret sharing scheme	
Dealer	Participants $p_1, p_2, \dots, p_n$
<p>Calculate <math>m = \binom{n}{t-1}</math>.            Choose abstract free generating set <math>X := \{x_1, x_2, \dots, x_m\}</math> and explicit free generating set <math>M := \{M_1, M_2, \dots, M_m\}</math> with <math>M_i \in \text{SL}(2, \mathbb{Q})</math> (all or almost all <math>M_i \notin \text{SL}(2, \mathbb{Z})</math>).</p> <p>Apply simultaneously regular Nielsen transformation (NT) on <math>X</math> and <math>M</math>:</p> $\begin{array}{ccc} (x_1, x_2, \dots, x_m) & & (M_1, M_2, \dots, M_m) \\ \downarrow \text{NT} & & \downarrow \text{NT} \\ (u_1, u_2, \dots, u_m) & & (N_1, N_2, \dots, N_m) \end{array}$ <p><math>U := \{u_1, u_2, \dots, u_m\}</math>; <math>N := \{N_1, N_2, \dots, N_m\}</math>.</p> <p>Construct sets <math>R_j \subseteq U</math> and <math>S_j \subseteq N</math> with share distribution method of Section 5.1.1;            it is <math> R_j  =  S_j  = \binom{n-1}{t-1}</math> for <math>j = 1, 2, \dots, n</math>.</p> <p>Distribute shares to the participants.</p>	<div style="text-align: center;"> <math>(R_1, S_1) \longrightarrow p_1</math>  <math>(R_2, S_2) \longrightarrow p_2</math>  <math>\vdots</math>  <math>(R_n, S_n) \longrightarrow p_n</math> </div> <p><math>t</math> participants combine their shares and thus get the sets <math>U</math> and <math>N</math>.</p> <p>Apply simultaneously regular Nielsen transformation (NT) on <math>U</math> and <math>N</math>:</p> $\begin{array}{ccc} (u_1, u_2, \dots, u_m) & & (N_1, N_2, \dots, N_m) \\ \downarrow \text{NT} & & \downarrow \text{NT} \\ (x'_1, x'_2, \dots, x'_m) & & (M'_1, M'_2, \dots, M'_m) \end{array}$ <p>The secret is</p> $S := \sum_{j=1}^m \frac{1}{ a'_j } \in \mathbb{Q}^+, \text{ with } a'_j := \text{tr}(M'_j) \in \mathbb{Q}.$

**Security 6.1.2.** For the calculation of the secret, the participants need the whole set  $M'$ , because the secret depends on the traces of the matrices  $M'_i \in M'$ . The participants need both sets  $U$  and  $N$ . If they just have one set  $U$  or  $N$  they cannot get information about the set  $M'$ . If the set  $U$  is known, it is only known which regular Nielsen transformation should be done to get the Nielsen equivalent set  $X'$ , but it is unknown on which matrices they should be simultaneously done.

If only the set  $N$  is known, then the matrices in  $\mathrm{SL}(2, \mathbb{Q})$  are known, but nobody knows which Nielsen transformations should be applied on  $N$  to get the set  $M'$ . It is also unknown how many Nielsen transformations were used. There could be hints, for Nielsen transformations if elements in  $N$  could be written in terms of other elements in  $\mathrm{SL}(2, \mathbb{Q})$ . For example it is known, that an algorithm exists which writes each element in  $\mathrm{SL}(2, \mathbb{Z})$  in terms of

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

(see Remark 4.3.12). We can use the fact that there is no algorithm known to solve the (constructive) membership problem for (discrete) free subgroups of  $\mathrm{SL}(2, \mathbb{Q})$  of rank 2 or greater than 2 which are not subgroups in  $\mathrm{SL}(2, \mathbb{Z})$  (see Remark 4.3.14). B. Eick, M. Kirschmer and C. Leedham-Green presented in their paper [EKL14] a practical algorithm to solve the constructive membership problem for discrete free subgroups of rank 2 of  $\mathrm{SL}(2, \mathbb{R})$ . For example, the subgroup  $\mathrm{SL}(2, \mathbb{Z})$  of  $\mathrm{SL}(2, \mathbb{R})$  is discrete, but they also mention, that it is an open problem to solve the membership problem for (discrete) free subgroups of  $\mathrm{SL}(2, \mathbb{R})$  with arbitrary rank  $m \geq 2$ . Therefore, if the dealer takes care, that all (or almost all) matrices in  $M$  are in  $\mathrm{SL}(2, \mathbb{Q})$  but not in  $\mathrm{SL}(2, \mathbb{Z})$  then the constructive membership problem cannot be used to get information about the Nielsen transformation to go from  $N$  to  $M'$ .

Running time:

In [Ste89] an algorithm, using elementary Nielsen transformations, is presented which, given a finite set  $S$  of  $m$  words of a free group, returns a set  $S'$  of Nielsen reduced words, such that  $\langle S \rangle = \langle S' \rangle$ ; the algorithm runs in  $\mathcal{O}(\ell^2 m^2)$ , where  $\ell$  is the maximum free length of a word in  $S$ . In this cryptographic protocol, the dealer fixes the number  $m$ , hence the running time depends only on the maximum free length  $\ell$  of the words in the Nielsen equivalent set  $U$  to the set  $X$ . Thus, the participants have a running time of  $\mathcal{O}(\ell^2)$  to get the set  $X'$ .

If the participants perform the associated elementary Nielsen transformations on the set  $N$  of matrices at the same time, then they perform either a matrix multiplication or they calculate an inverse matrix. In order to multiply two  $2 \times 2$  matrices in  $\mathrm{SL}(2, \mathbb{Q})$  they need 8 multiplications and 4 additions of rational numbers, hence 12 operations. The inverse matrix of

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}(2, \mathbb{Q})$$

is

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

The participants need 4 operations to get  $\frac{1}{ad - bc}$  (note, that in  $\mathrm{SL}(2, \mathbb{Q})$  it is  $ad - bc = 1$  and hence  $A^{-1} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$ ); for the entries of the matrix  $A^{-1}$  they do not need any operations, they just swap two entries and write a minus in front of the other two entries.

All together the participants have a running time of  $\mathcal{O}(\ell^2)$ , where  $\ell$  is the maximum free length of the elements in  $U$ .

We now present an example for this secret sharing scheme.

**Example 6.1.3.** We perform the steps for a  $(3, 2)$ -secret sharing scheme with the help of the computer program Maple 16, see Appendix C.3. It is  $n = 3$ ,  $t = 2$  and hence  $m = \binom{3}{1} = 3$ . First the dealer generates the shares for the participants.

1. The dealer chooses an abstract presentation for the free group  $F$  of rank 3

$$F = \langle X \mid \ \rangle \quad \text{with } X := \{x_1, x_2, x_3\}.$$

He takes an explicit presentation

$$F = \langle M \mid \ \rangle \quad \text{with } M := \{M_1, M_2, M_3\},$$

$M_i \in \text{SL}(2, \mathbb{Q})$ , with the help of Theorem 4.2.18. We first mention that the inequalities (4.1) hold for

$$r_1 = \frac{7}{2}, \quad r_2 = \frac{15}{2}, \quad r_3 = 11$$

and hence the set of the matrices

$$\begin{aligned} M_1 &= \begin{pmatrix} -\frac{7}{2} & -1 + \left(\frac{7}{2}\right)^2 \\ 1 & -\frac{7}{2} \end{pmatrix} = \begin{pmatrix} -\frac{7}{2} & \frac{45}{4} \\ 1 & -\frac{7}{2} \end{pmatrix}, \\ M_2 &= \begin{pmatrix} -\frac{15}{2} & -1 + \left(\frac{15}{2}\right)^2 \\ 1 & -\frac{15}{2} \end{pmatrix} = \begin{pmatrix} -\frac{15}{2} & \frac{221}{4} \\ 1 & -\frac{15}{2} \end{pmatrix} \text{ and} \\ M_3 &= \begin{pmatrix} -11 & -1 + 11^2 \\ 1 & -11 \end{pmatrix} = \begin{pmatrix} -11 & 120 \\ 1 & -11 \end{pmatrix} \end{aligned}$$

is a free generating set for a free group of rank 3.

2. It is

$$a_1 := \text{tr}(M_1) = -7, \quad a_2 := \text{tr}(M_2) = -15, \quad a_3 := \text{tr}(M_3) = -22,$$

and hence the secret is

$$S := \sum_{j=1}^3 \frac{1}{|a_j|} = \frac{589}{2310}.$$

3. Construction of the shares for the participants: First the dealer applies regular Nielsen transformations (NTs) simultaneously for both sets  $X$  and  $M$  to get Nielsen equivalent sets  $U$  and  $N$  to  $X$  and  $M$ , respectively. These transformations are shown in Table 6.2 (page 141).

Table 6.2.: Nielsen transformations (NTs) done by the dealer

NTs	Theoretical set X	Explicit set M
	$(x_1, x_2, x_3)$	$\left( \left( \begin{smallmatrix} -\frac{7}{2} & 45 \\ 1 & -\frac{7}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -\frac{15}{2} & 221 \\ 1 & -\frac{15}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -11 & 120 \\ 1 & -11 \end{smallmatrix} \right) \right)$
$(T1)_2$	$(x_1, x_2^{-1}, x_3)$	$\left( \left( \begin{smallmatrix} -\frac{7}{2} & 45 \\ 1 & -\frac{7}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -\frac{15}{2} & -221 \\ -1 & -\frac{15}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -11 & 120 \\ 1 & -11 \end{smallmatrix} \right) \right)$
$(T2)_{1.2}$	$(x_1 x_2^{-1}, x_2^{-1}, x_3)$	$\left( \left( \begin{smallmatrix} 15 & 109 \\ -4 & -29 \end{smallmatrix} \right), \left( \begin{smallmatrix} -\frac{15}{2} & -221 \\ -1 & -\frac{15}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -11 & 120 \\ 1 & -11 \end{smallmatrix} \right) \right)$
$[(T2)_{3.2}]^3$	$(x_1 x_2^{-1}, x_2^{-1}, x_3 x_2^{-3})$	$\left( \left( \begin{smallmatrix} 15 & 109 \\ -4 & -29 \end{smallmatrix} \right), \left( \begin{smallmatrix} -\frac{15}{2} & -221 \\ -1 & -\frac{15}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -8565 & -63664 \\ 799 & 5939 \end{smallmatrix} \right) \right)$
$(T2)_{2.3}$	$(x_1 x_2^{-1}, x_2^{-1} x_3 x_2^{-3}, x_3 x_2^{-3})$	$\left( \left( \begin{smallmatrix} 15 & 109 \\ -4 & -29 \end{smallmatrix} \right), \left( \begin{smallmatrix} \frac{80371}{2} & \frac{597401}{2} \\ \frac{5145}{2} & \frac{38243}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -8565 & -63664 \\ 799 & 5939 \end{smallmatrix} \right) \right)$
$(T1)_1$	$(x_2 x_1^{-1}, x_2^{-1} x_3 x_2^{-3}, x_3 x_2^{-3})$	$\left( \left( \begin{smallmatrix} -29 & -109 \\ 4 & 15 \end{smallmatrix} \right), \left( \begin{smallmatrix} \frac{80371}{2} & \frac{597401}{2} \\ \frac{5145}{2} & \frac{38243}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -8565 & -63664 \\ 799 & 5939 \end{smallmatrix} \right) \right)$
$(T2)_{1.2}$	$(x_2 x_1^{-1} x_2^{-1} x_3 x_2^{-3}, x_2^{-1} x_3 x_2^{-3}, x_3 x_2^{-3})$	$\left( \left( \begin{smallmatrix} -\frac{3452369}{2} & -\frac{25661603}{2} \\ \frac{237917}{2} & \frac{1768447}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} \frac{80371}{2} & \frac{597401}{2} \\ \frac{5145}{2} & \frac{38243}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} -8565 & -63664 \\ 799 & 5939 \end{smallmatrix} \right) \right)$
$(T1)_3$	$(x_2 x_1^{-1} x_2^{-1} x_3 x_2^{-3}, x_2^{-1} x_3 x_2^{-3}, x_2^3 x_3^{-1})$	$\left( \left( \begin{smallmatrix} -\frac{3452369}{2} & -\frac{25661603}{2} \\ \frac{237917}{2} & \frac{1768447}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} \frac{80371}{2} & \frac{597401}{2} \\ \frac{5145}{2} & \frac{38243}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} 5939 & 63664 \\ -799 & -8565 \end{smallmatrix} \right) \right)$
$(T2)_{3.2}$	$(x_2 x_1^{-1} x_2^{-1} x_3 x_2^{-3}, x_2^{-1} x_3 x_2^{-3}, x_2^3 x_3^{-1} x_2^{-1} x_3 x_2^{-3})$	$\left( \left( \begin{smallmatrix} -\frac{3452369}{2} & -\frac{25661603}{2} \\ \frac{237917}{2} & \frac{1768447}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} \frac{80371}{2} & \frac{597401}{2} \\ \frac{5145}{2} & \frac{38243}{2} \end{smallmatrix} \right), \left( \begin{smallmatrix} \frac{1132425929}{4} & \frac{8417369243}{4} \\ -\frac{152350279}{4} & -\frac{1132425989}{4} \end{smallmatrix} \right) \right)$

The dealer gets the sets

$$U = \{u_1, u_2, u_3\} := \{x_2 x_1^{-1} x_2^{-1} x_3 x_2^{-3}, x_2^{-1} x_3 x_2^{-3}, x_2^3 x_3^{-1} x_2^{-1} x_3 x_2^{-3}\}$$

and

$$N = \{N_1, N_2, N_3\} \\ := \left\{ \left( -\frac{3452369}{2}, -\frac{25661603}{2} \right), \left( \frac{80371}{2}, \frac{597401}{2} \right), \left( \frac{1132425929}{4}, \frac{8417369243}{4} \right) \right\}.$$

He gets the share  $(R_i, S_j)$  for the participant  $p_k$  with  $R_i \subset U$  and  $S_j \subset N$  following the method in Section 5.1.1:

- a) It is  $m = \binom{n}{t-1} = \binom{3}{1} = 3$ .
- b) The dealer chooses the abstract elements  $\tilde{a}_1, \tilde{a}_2, \tilde{a}_3$  and gets the three sets

$$A_1 = \{1\}, \quad A_2 = \{2\}, \quad A_3 = \{3\}.$$

With the help of the  $A_i$  the dealer gets the sets  $R'_1, R'_2$ , and  $R'_3$  which contain elements from the set  $\{\tilde{a}_1, \tilde{a}_2, \tilde{a}_3\}$ . He puts the element  $\tilde{a}_j$  by which  $i$  is not contained in the set  $A_j$  for  $i = 1, 2, 3$  and  $j = 1, 2, 3$ , into the set  $R'_i$ :

$$\begin{aligned} 1 \notin A_2, A_3 &\implies R'_1 = \{\tilde{a}_2, \tilde{a}_3\}, \\ 2 \notin A_1, A_3 &\implies R'_2 = \{\tilde{a}_1, \tilde{a}_3\}, \\ 3 \notin A_1, A_2 &\implies R'_3 = \{\tilde{a}_1, \tilde{a}_2\}. \end{aligned}$$

Now, we apply this to  $U$  and  $N$  to create the share-sets for the participants, respectively:

$$\begin{aligned} R_1 &= \{u_2, u_3\}, & S_1 &= \{N_2, N_3\}, \\ R_2 &= \{u_1, u_3\}, & S_2 &= \{N_1, N_3\}, \\ R_3 &= \{u_1, u_2\}, & S_3 &= \{N_1, N_2\}. \end{aligned}$$

4. The dealer gives each participant  $p_k$  a tuple  $(R_i, S_j)$ . Participant  $p_1$  gets  $(R_1, S_2)$ ,  $p_2$  gets  $(R_2, S_3)$  and  $p_3$  gets  $(R_3, S_1)$ .

Assume the participants  $p_1$  and  $p_2$  come together to reconstruct the secret. They generate the sets  $U = \{u_1, u_2, u_3\}$  and  $N = \{N_1, N_2, N_3\}$ . The secret can be recovered as follow.

The participants apply regular Nielsen transformations step by step simultaneously for both sets  $U$  and  $N$  to get  $X'$  and  $M'$ . The steps are shown in Table 6.3 (page 143) and Table 6.4 (page 144).

Table 6.3.: Nielsen transformations (NTs) done by the participants I

NTs	Theoretical set U	Explicit set N
	$(x_2x_1^{-1}x_2^{-1}x_3x_2^{-3},$ $x_2^{-1}x_3x_2^{-3}, x_2^3x_3^{-1}x_2^{-1}x_3x_2^{-3})$	$\left( \left( \begin{pmatrix} -3452369 & -25661603 \\ 237917 & 1768447 \end{pmatrix}, \right. \right.$ $\left. \left( \begin{pmatrix} 80371 & 597401 \\ 5145 & 38243 \end{pmatrix}, \begin{pmatrix} 1132425929 & 8417369243 \\ -152350279 & -1132425989 \end{pmatrix} \right) \right)$
$(T1)_2$	$(x_2x_1^{-1}x_2^{-1}x_3x_2^{-3},$ $x_2^3x_3^{-1}x_2, x_2^3x_3^{-1}x_2^{-1}x_3x_2^{-3})$	$\left( \left( \begin{pmatrix} -3452369 & -25661603 \\ 237917 & 1768447 \end{pmatrix}, \right. \right.$ $\left. \left( \begin{pmatrix} 38243 & -597401 \\ -5145 & 80371 \end{pmatrix}, \begin{pmatrix} 1132425929 & 8417369243 \\ -152350279 & -1132425989 \end{pmatrix} \right) \right)$
$(T2)_{3.2}$	$(x_2x_1^{-1}x_2^{-1}x_3x_2^{-3},$ $x_2^3x_3^{-1}x_2, x_2^3x_3^{-1})$	$\left( \left( \begin{pmatrix} -3452369 & -25661603 \\ 237917 & 1768447 \end{pmatrix}, \right. \right.$ $\left. \left( \begin{pmatrix} 38243 & -597401 \\ -5145 & 80371 \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right) \right)$
$(T1)_2$	$(x_2x_1^{-1}x_2^{-1}x_3x_2^{-3},$ $x_2^{-1}x_3x_2^{-3}, x_2^3x_3^{-1})$	$\left( \left( \begin{pmatrix} -3452369 & -25661603 \\ 237917 & 1768447 \end{pmatrix}, \right. \right.$ $\left. \left( \begin{pmatrix} 80371 & 597401 \\ 5145 & 38243 \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right) \right)$
$(T2)_{2.3}$	$(x_2x_1^{-1}x_2^{-1}x_3x_2^{-3},$ $x_2^{-1}, x_2^3x_3^{-1})$	$\left( \left( \begin{pmatrix} -3452369 & -25661603 \\ 237917 & 1768447 \end{pmatrix}, \right. \right.$ $\left. \left( \begin{pmatrix} -15 & 221 \\ -1 & -15 \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right) \right)$
$(T2)_{1.3}$	$(x_2x_1^{-1}x_2^{-1}, x_2^{-1}, x_2^3x_3^{-1})$	$\left( \left( \begin{pmatrix} 653 & 9679 \\ -45 & -667 \end{pmatrix}, \begin{pmatrix} -15 & 221 \\ -1 & -15 \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right) \right)$
$(T1)_2$	$(x_2x_1^{-1}x_2^{-1}, x_2, x_2^3x_3^{-1})$	$\left( \left( \begin{pmatrix} 653 & 9679 \\ -45 & -667 \end{pmatrix}, \begin{pmatrix} -15 & 221 \\ 1 & -15 \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right) \right)$
$(T2)_{1.2}$	$(x_2x_1^{-1}, x_2, x_2^3x_3^{-1})$	$\left( \left( \begin{pmatrix} -29 & -109 \\ 4 & 15 \end{pmatrix}, \begin{pmatrix} -15 & 221 \\ 1 & -15 \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right) \right)$
$(T1)_1$	$(x_1x_2^{-1}, x_2, x_2^3x_3^{-1})$	$\left( \left( \begin{pmatrix} 15 & 109 \\ -4 & -29 \end{pmatrix}, \begin{pmatrix} -15 & 221 \\ 1 & -15 \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right) \right)$

Table 6.4.: Nielsen transformations (NTs) done by the participants II

NTs	Theoretical set U	Explicit set N
$(T2)_{1,2}$	$(x_1, x_2, x_2^3 x_3^{-1})$	$\left( \begin{pmatrix} -\frac{7}{2} & 45 \\ 1 & -\frac{7}{2} \end{pmatrix}, \begin{pmatrix} -\frac{15}{2} & 221 \\ 1 & -\frac{4}{15} \end{pmatrix}, \begin{pmatrix} 5939 & 63664 \\ -799 & -8565 \end{pmatrix} \right)$
$(T1)_3$	$(x_1, x_2, x_3 x_2^{-3})$	$\left( \begin{pmatrix} -\frac{7}{2} & 45 \\ 1 & -\frac{7}{2} \end{pmatrix}, \begin{pmatrix} -\frac{15}{2} & 221 \\ 1 & -\frac{4}{15} \end{pmatrix}, \begin{pmatrix} -8565 & -63664 \\ 799 & 5939 \end{pmatrix} \right)$
$[(T2)_{3,2}]^3$	$(x_1, x_2, x_3)$	$\left( \begin{pmatrix} -\frac{7}{2} & 45 \\ 1 & -\frac{7}{2} \end{pmatrix}, \begin{pmatrix} -\frac{15}{2} & 221 \\ 1 & -\frac{4}{15} \end{pmatrix}, \begin{pmatrix} -11 & 120 \\ 1 & -11 \end{pmatrix} \right)$

With the knowledge of the set  $M' = \left\{ \begin{pmatrix} -\frac{7}{2} & 45 \\ 1 & -\frac{7}{2} \end{pmatrix}, \begin{pmatrix} -\frac{15}{2} & 221 \\ 1 & -\frac{4}{15} \end{pmatrix}, \begin{pmatrix} -11 & 120 \\ 1 & -11 \end{pmatrix} \right\}$  the participants can reconstruct the secret easily. It is

$$a'_1 := \text{tr}(M'_1) = -7, \quad a'_2 := \text{tr}(M'_2) = -15, \quad a'_3 := \text{tr}(M'_3) = -22,$$

and hence it is

$$S := \sum_{j=1}^3 \frac{1}{|a'_j|} = \frac{1}{7} + \frac{1}{15} + \frac{1}{22} = \frac{589}{2310}.$$

In general we can use any free matrix group  $F$  of rank  $m := \binom{n}{t-1}$  for a  $(n, t)$ -secret sharing scheme as it is described in this example. The shares can be generated by the method of D. Panagopoulos and are tuples  $(R_i, S_j)$  with  $R_i \subset U$  and  $S_j \subset N$ .

We could also use some other sums or products to define a secret. Some examples are:

- $S := \prod_{i=1}^m |\text{tr}(M_i)|$  or  $S := \sum_{i=1}^m |\text{tr}(M_i)|$  or
- $S := \prod_{i=1}^m (\text{tr}(M_i))^2$  or  $S := \sum_{i=1}^m (\text{tr}(M_i))^2$  or
- $S := \prod_{i=1}^{\frac{m}{2}} \text{tr}([M_{2i-1}, M_{2i}])$  if  $m$  is even or
- $S := \sum_{i=1}^m \text{tr}(M_i^2)$ .

**Remark 6.1.4.** Maybe the dealer should not only use matrices, for the set  $M$ , generated by Theorem 4.2.18, because they are of a special look, which is

$$M_j = \begin{pmatrix} -r_j & -1 + r_j^2 \\ 1 & -r_j \end{pmatrix},$$



with  $r_j \in \mathbb{Q}$ ,  $1 \leq j \leq m$ , and it is

$$r_{j+1} - r_j \geq 3 \quad \text{and} \quad r_1 \geq 2.$$

Maybe an enemy could start an attack with this information. To avoid this, the dealer could calculate a set  $X' = \{X_1, X_2, \dots, X_\ell\}$ ,  $\ell \geq 3$ , with matrices generated by Theorem 4.2.18, and the set  $M$  could be a free generating set (not necessary Nielsen reduced) with  $m = \binom{n}{t-1}$  elements of a subgroup of  $\langle X' \mid \quad \rangle$ .

**Remark 6.1.5.** We now compare this scheme as  $(n, t)$ -secret sharing protocol to Shamir's properties as given in Section 5.4. To get a  $(n, t)$ -secret sharing variation we use the share distribution method given in Section 5.1.1 by D. Panagopoulos. Therefore, this scheme fulfills the same properties of Shamir as D. Panagopoulos' scheme does. That means (3) and (4) are fulfilled and (1) and (2) are not fulfilled. Furthermore, the additional property (5) does not hold. In more details:

- (1) The size of each share exceed the size of the secret, because the secret is a rational number and the shares are tuples of subsets  $(R_i, S_j)$ , whereby each set  $R_i$  and  $S_j$  holds  $\binom{n-1}{t-1}$  elements (see Section 5.1.1).
- (2) The dealer creates the shares according to instructions (see for instance the summary in Section 5.1.1). Hence, he cannot add or delete shares, because the way he creates them depends on the number  $m = \binom{n}{t-1}$  and the number  $n$  of participants.
- (3) He can change the shares if he changes the set  $M$ . He has to take care that the new set  $M^{new}$  (which is another explicit generating set for a free group of rank  $m$ ) gives the same secret as the set  $M$ , that means

$$\sum_{i=1}^m \frac{1}{|a_i|} = \sum_{i=1}^m \frac{1}{|a_i^{new}|},$$

with  $a_i := \text{tr}(M_i)$ ,  $M_i \in M$ , and  $a_i^{new} := \text{tr}(M_i^{new})$ ,  $M_i^{new} \in M^{new}$ .

- (4) Every  $(n, t)$ -secret sharing scheme can be converted into a hierarchical secret sharing protocol (see Section 5.3 Remark 5.3.10).
- (5) A secret cannot be changed easily without changing the shares, because it is a sum over the rank of matrices in the set  $M$  and hence depends on this set.

To fulfill property (5) we introduce the following variation.

**Variation 6.1.6.** The dealer constructs the Nielsen equivalent sets  $U$  to  $X$  and  $N$  to  $M$  as in Figure 6.1. The set  $U$  is distributed to the participants and the set  $N$  is public. If the dealer needs a new secret he uses a new set  $\tilde{M}$  and calculates, with the same Nielsen transformation with which he comes from the set  $X$  to the set  $U$ , the Nielsen equivalent set  $\tilde{N}$  to  $\tilde{M}$ . He then publishes the set  $\tilde{N}$ . If the participants come together to reconstruct the set  $U$ , they know the Nielsen transformation to come from the set  $N$  to the set  $M$  or from the set  $\tilde{N}$  to the set  $\tilde{M}$ , respectively. The dealer can change the secret without changing the shares of the participants. If he takes care, that all (or almost all) matrices in the sets  $M$  or  $\tilde{M}$ , respectively, are in  $\text{SL}(2, \mathbb{Q})$  but not in  $\text{SL}(2, \mathbb{Z})$  then the constructive membership problem cannot be used to get information about the used Nielsen transformation, see Security 6.1.2.

## 6.2. Secret sharing scheme based on Nielsen reduced sets and the free length (Protocol 7)

We now present **Protocol 7**, which uses as **Protocol 6** Nielsen transformations. Let  $F$  be a finitely generated free group with the abstract free generating set  $X := \{x_1, x_2, \dots, x_q\}$ ,  $q \in \mathbb{N} \setminus \{1\}$ , that is,

$$F = \langle X \mid \ \rangle.$$

In this cryptographic protocol we just work with respect to the given basis elements of a finitely generated free group.

Steps for the dealer:

For a  $(m, m)$ -secret sharing scheme,  $m \in \mathbb{N}$ , the dealer does the following steps.

1. He chooses a free group  $F$  with an abstract free generating set  $X := \{x_1, x_2, \dots, x_q\}$ ,  $q \in \mathbb{N} \setminus \{1\}$ , and a Nielsen reduced set  $U \subset F$  with  $U := \{u_1, u_2, \dots, u_m\}$ . The  $u_i$  are given as words in  $X$ .
2. With the known set  $U$  the secret is the sum

$$S := \sum_{i=1}^m \frac{1}{|u_i|_X},$$

with  $|u_i|_X$  the free length of the word  $u_i$  in  $X$ . If the dealer needs a special secret he can act as described in Variation 5.2.2 or Variation 5.2.4.

3. To generate the shares for the participants, the dealer does a regular Nielsen transformation on the set  $U$  to get the Nielsen equivalent set  $V$  as shown in Figure 6.3.

$$\begin{array}{c} U = \{u_1, u_2, \dots, u_m\} \\ (u_1, u_2, \dots, u_m) \\ \downarrow \text{regular Nielsen} \\ \text{transformation} \\ (v_1, v_2, \dots, v_m) \\ V = \{v_1, v_2, \dots, v_m\} \end{array}$$

Figure 6.3.: Regular Nielsen transformation from  $U$  to a Nielsen equivalent set  $V$

4. Let  $P = \{p_1, p_2, \dots, p_m\}$  be the set of the participants for the secret sharing scheme. Each participant  $p_i$ ,  $1 \leq i \leq m$ , gets one element  $v_i \in V$ .

If all participants come together to reconstruct the secret, they combine their shares and get the set  $V = \{v_1, v_2, \dots, v_m\}$ .

Steps for the participants:

1. They have to find a Nielsen reduced set  $U' = \{u'_1, u'_2, \dots, u'_m\}$  to  $V$ . They apply Nielsen transformations in a Nielsen reducing manner as described in [Ste89], Remark 4.2.17, (or also in [CgRR08] and [LS77]) and get from  $V$  a Nielsen reduced set  $U'$ , see Figure 6.4.

$$\begin{array}{c}
 V = \{v_1, v_2, \dots, v_m\} \\
 (v_1, v_2, \dots, v_m) \\
 \downarrow \text{regular Nielsen} \\
 \text{transformation} \\
 \downarrow \\
 (u'_1, u'_2, \dots, u'_m) \\
 U' = \{u'_1, u'_2, \dots, u'_m\}
 \end{array}$$

Figure 6.4.: Regular Nielsen transformation from  $V$  to a Nielsen reduced set  $U'$

2. With the knowledge of  $U'$  the secret is the sum

$$S = \sum_{i=1}^m \frac{1}{|u'_i|_X},$$

because for each  $i$  we have  $|u'_i|_X = |u_j|_X$  for some  $j$  (see the proof of Corollary 3.1 in [MKS66]). From  $U'$  we get  $U$  by permutations and length preserving Nielsen transformations.

**Remark 6.2.1.** If a  $(n, t)$ -secret sharing scheme is desired the dealer chooses  $m = \binom{n}{t-1}$  and uses the method described in Section 5.1.1 for distribution of  $V$  and gives one of the obtained share-sets, that is,  $R_i \subseteq V$ , to participant  $p_i$ ,  $1 \leq i \leq n$ . If  $t$  or more participants combine their shares, they get the set  $V$ . With this set they are able to reconstruct the secret as explained above in the steps for the participants. Less than  $t$  participants cannot get the whole set  $V$  due to the method how the share-sets  $R_i \subseteq V$  are generated.

This cryptographic protocol as a  $(n, t)$ -secret sharing scheme is summarized in Table 6.5 (page 148).

Table 6.5.: Summary of **Protocol 7**: Secret sharing scheme using Nielsen transformations together with Nielsen reduced sets and free lengths of certain words

( $n, t$ )-secret sharing scheme	
Dealer	Participants $p_1, p_2, \dots, p_n$
<p>Calculate <math>m = \binom{n}{t-1}</math>.            Choose abstract free generating set <math>X = \{x_1, x_2, \dots, x_q\}</math>, <math>q \in \mathbb{N} \setminus \{1\}</math>, and a Nielsen reduced set <math>U = \{u_1, u_2, \dots, u_m\} \subset F</math>, <math>u_i</math> words in <math>X</math>.</p> <p>Apply regular Nielsen transformation (NT) on <math>U</math>:</p> $\begin{array}{c} (u_1, u_2, \dots, u_m) \\ \downarrow \text{NT} \\ (v_1, v_2, \dots, v_m) \end{array}$ <p><math>V := \{v_1, v_2, \dots, v_m\}</math>.</p> <p>Construct sets <math>R_j \subseteq V</math> with share distribution method of Section 5.1.1;            it is <math> R_j  = \binom{n-1}{t-1}</math> for <math>j = 1, 2, \dots, n</math>.</p> <p>Distribute shares to the participants.</p>	<div style="text-align: center;"> <math>\xrightarrow{R_1} p_1</math>  <math>\xrightarrow{R_2} p_2</math>  <math>\vdots</math>  <math>\xrightarrow{R_n} p_n</math> </div> <p><math>t</math> participants combine their shares and thus get the set <math>V</math>.</p> <p>Apply regular Nielsen transformation (NT) on <math>V</math> to get a Nielsen reduced set:</p> $\begin{array}{c} (v_1, v_2, \dots, v_m) \\ \downarrow \text{NT} \\ (u'_1, u'_2, \dots, u'_m) \end{array}$ <p>The secret is</p> $S = \sum_{i=1}^m \frac{1}{ u'_i _X}.$

**Security 6.2.2.** By combining less than  $m$  shares the participants get a subset  $\tilde{V}$  of  $V$ , it is

$|\tilde{V}| \leq m - 1$ . If they apply a regular Nielsen transformation on the set  $\tilde{V}$  in a Nielsen reducing manner they do not get a subset  $\tilde{U}$  of  $U$ , in general. Hence, they get no useful information to reconstruct the secret. They know that for each element  $\tilde{u}_i \in \tilde{U}$  there exists an element  $u_i \in U$ , such that  $|u_i|_X \leq |\tilde{u}_i|_X$ . Hence, they only know, that  $S > \sum_{\tilde{u}_i \in \tilde{U}} \frac{1}{|\tilde{u}_i|_X}$ , because  $\frac{1}{x} \leq \frac{1}{y}$  if  $0 < y \leq x$ .

If the secret is just the sum  $\sum_{i=1}^m |u_i|_X$ , then the participants know that the secret is less than the sum over the free length of the elements in  $V$ , that is,  $S < \sum_{i=1}^m |v_i|_X$ , because it is known that a Nielsen reduced set  $V$  has the shortest total  $X$ -length of all Nielsen equivalent sets to this set  $V$ , see Proposition 4.2.6. It is likely that the secret is less than

$$K = \sum_{\substack{i=1 \\ i \neq j}}^m |v_i|_X$$

for a  $j \in \{1, 2, \dots, m\}$ . Hence, less than  $m$  participants could get a correct bound  $K$  for the secret, which is a natural number. Therefore, they know that the secret  $S$  is an element in  $\{m, m + 1, m + 2, \dots, K\}$ .

Instead, as mention in Security 6.2.2, if the secret is  $S = \sum_{i=1}^m \frac{1}{|u_i|_X}$  and  $m - 1$  participants calculate the sum

$$S' = \sum_{\substack{i=1 \\ i \neq j}}^m \frac{1}{|v_i|_X},$$

they can only assume  $S > S'$ , because there is a bijection between  $V$  and  $U$ , such that  $v_i \mapsto u_j$  and  $|v_i|_X \geq |u_j|_X$  for each  $i$  and the corresponding  $j$  and in addition it is  $\frac{1}{x} \leq \frac{1}{y}$  if  $0 < y \leq x$ .

### Running time:

In [Ste89] an algorithm, using elementary Nielsen transformations, is presented which, given a finite set  $S$  of  $m$  words of a free group, returns a set  $S'$  of Nielsen reduced words, such that  $\langle S \rangle = \langle S' \rangle$ ; the algorithm runs in  $\mathcal{O}(\ell^2 m^2)$ , where  $\ell$  is the maximum free length of a word in  $S$ . In this cryptographic protocol, the dealer fixes the number  $m$ , hence the running time depends only on the maximum free length  $\ell$  of the words in the Nielsen equivalent set  $V$  to the set  $U$ . Thus, the participants have a running time of  $\mathcal{O}(\ell^2)$  to get the set  $U'$ . The secret is then the above sum, which is computable in linear time.

We now present an example for this secret sharing scheme.

**Example 6.2.3.** This example is executed with the help of the computer program GAP<sup>1</sup> and the FGA<sup>2</sup> package. We give a (3, 3)-secret sharing example. Let  $p_1, p_2$  and  $p_3$  be the participants. Let  $F$  be a finitely generated free group with the free generating set  $X = \{a, b, c\}$ . The Nielsen reduced set  $U$  can be constructed with the help of Lemma 4.2.15 and because of Theorem 4.2.13 it is a basis for a free group. The dealer chooses  $U = \{b^2 a, cab, ac^{-1} b^{-1} a^3\}$ , with  $u_1 = b^2 a$ ,  $u_2 = cab$  and  $u_3 = ac^{-1} b^{-1} a^3$ . Thus, the secret is

$$S = \sum_{i=1}^3 \frac{1}{|u_i|_X} = \frac{1}{3} + \frac{1}{3} + \frac{1}{6} = \frac{5}{6}.$$

<sup>1</sup>Groups, Algorithms and Programming [GAP15]

<sup>2</sup>Free Group Algorithms. A GAP4 package by Christian Sievers, TU Braunschweig.

The regular Nielsen transformation for the dealer is given by the elementary Nielsen transformations

$$(N1)_2 (N2)_{1.3} (N2)_{3.2} (N2)_{2.3} (N1)_2 (N2)_{1.2} (N1)_2 (N2)_{3.2} (N1)_1 (N1)_{3.1} (N1)_1,$$

which are applied from the left to the right:

$$\begin{aligned} (u_1, u_2, u_3) &\xrightarrow{(N1)_2} (u_1, u_2^{-1}, u_3) \xrightarrow{(N2)_{1.3}} (u_1 u_3, u_2^{-1}, u_3) \xrightarrow{(N2)_{3.2}} (u_1 u_3, u_2^{-1}, u_3 u_2^{-1}) \\ &\xrightarrow{(N2)_{2.3}} (u_1 u_3, u_2^{-1} u_3 u_2^{-1}, u_3 u_2^{-1}) \xrightarrow{(N1)_2} (u_1 u_3, u_2 u_3^{-1} u_2, u_3 u_2^{-1}) \\ &\xrightarrow{(N2)_{1.2}} (u_1 u_3 u_2 u_3^{-1} u_2, u_2 u_3^{-1} u_2, u_3 u_2^{-1}) \xrightarrow{(N1)_2} (u_1 u_3 u_2 u_3^{-1} u_2, u_2^{-1} u_3 u_2^{-1}, u_3 u_2^{-1}) \\ &\xrightarrow{(N2)_{3.2}} (u_1 u_3 u_2 u_3^{-1} u_2, u_2^{-1} u_3 u_2^{-1}, u_3 u_2^{-2} u_3 u_2^{-1}) \\ &\xrightarrow{(N1)_1} (u_2^{-1} u_3 u_2^{-1} u_3^{-1} u_1^{-1}, u_2^{-1} u_3 u_2^{-1}, u_3 u_2^{-2} u_3 u_2^{-1}) \\ &\xrightarrow{(N1)_{3.1}} (u_2^{-1} u_3 u_2^{-1} u_3^{-1} u_1^{-1}, u_2^{-1} u_3 u_2^{-1}, u_3 u_2^{-2} u_3 u_2^{-2} u_3 u_2^{-1} u_3^{-1} u_1^{-1}) \\ &\xrightarrow{(N1)_1} (u_1 u_3 u_2 u_3^{-1} u_2, u_2^{-1} u_3 u_2^{-1}, u_3 u_2^{-2} u_3 u_2^{-2} u_3 u_2^{-1} u_3^{-1} u_1^{-1}) \end{aligned}$$

This regular Nielsen transformation generates the automorphism  $f$  (see Proposition 4.2.4) with

$$\begin{aligned} f : \langle U \rangle &\rightarrow \langle U \rangle \\ u_1 &\mapsto v_1 := u_1 u_3 u_2 u_3^{-1} u_2 \\ u_2 &\mapsto v_2 := u_2^{-1} u_3 u_2^{-1} \\ u_3 &\mapsto v_3 := u_3 u_2^{-2} u_3 u_2^{-2} u_3 u_2^{-1} u_3^{-1} u_1^{-1}. \end{aligned}$$

Therefore, a Nielsen equivalent set of  $U$  is  $V = \{v_1, v_2, v_3\}$  with

$$\begin{aligned} v_1 &= b^2 a^2 c^{-1} b^{-1} a^3 c a b a^{-3} b c a^{-1} c a b, \\ v_2 &= b^{-1} a^{-1} c^{-1} a c^{-1} b^{-1} a^3 b^{-1} a^{-1} c^{-1}, \\ v_3 &= (a c^{-1} b^{-1} a^3 (b^{-1} a^{-1} c^{-1})^2 a c^{-1} b^{-1} a^3 b^{-1} a^{-1} c^{-1} a^{-3} b c a^{-2} b^{-2}). \end{aligned}$$

Each participant  $p_i$  gets one share  $v_i$ ,  $1 \leq i \leq 3$ .

If all three shares are combined, the participants regenerate the set  $V$  and if they now calculate a Nielsen reduced set of this set  $V$  they get a set  $U' := \{u'_1, u'_2, u'_3\}$  which includes two elements of free length 3 and one element of free length 6. Let  $|u'_1| = |u'_2| = 3$  and  $|u'_3| = 6$ , if this is not the situation, renumber the elements in  $U'$ . They reconstructed the correct secret

$$S = \sum_{i=1}^3 \frac{1}{|u'_i|_X} = \frac{1}{3} + \frac{1}{3} + \frac{1}{6} = \frac{5}{6}.$$

The package FGA for the computer program GAP provides the operation

▷ `FreeGeneratorsOfGroup(G)`

which returns a list of free Nielsen reduced generators, which defines a Nielsen reduced set of the finitely generated subgroup  $G$  of a free group. See Appendix C.4 for the computer code of this example.

**Remark 6.2.4.** We now compare this scheme as  $(n, t)$ -secret sharing protocol to Shamir's

properties as given in Section 5.4. To get a  $(n, t)$ -secret sharing variation we use the share distribution method given in Section 5.1.1 by D. Panagopoulos. Therefore, this scheme fulfills the same properties of Shamir as D. Panagopoulos' scheme does. That means (3) and (4) are fulfilled and (1) and (2) are not fulfilled. Furthermore, the additional property (5) does not hold. In more details:

- (1) The size of each share exceed the size of the secret, because the secret is a rational number and the shares  $R_i$  are subsets of  $V$ , whereby each set  $R_i$  holds  $\binom{n-1}{t-1}$  elements (see Section 5.1.1).
- (2) The dealer creates the shares according to instructions (see for instance the summary in Section 5.1.1). Hence, he cannot add or delete shares, because the way he creates them depends on the number  $m = \binom{n}{t-1}$  and the number  $n$  of participants.
- (3) He can change the shares if he changes the set  $U$ . Firstly, he chooses a new abstract free generating set  $X^{new} = \{x_1, x_2, \dots, x_{q'}\}$ , with  $q' \in \mathbb{N} \setminus \{1\}$  not necessary  $q' = q$ . Secondly, he selects a Nielsen reduced set  $U^{new} = \{u_1^{new}, u_2^{new}, \dots, u_m^{new}\}$ , with  $u_i^{new}$  word in  $X^{new}$ ,  $1 \leq i \leq m$ . He has to take care that the new set  $U^{new}$  gives the same secret as the set  $U$ , that means

$$\sum_{i=1}^m \frac{1}{|u_i|_X} = \sum_{i=1}^m \frac{1}{|u_i^{new}|_{X^{new}}},$$

with  $u_i \in U$  and  $u_i^{new} \in U^{new}$ .

- (4) Every  $(n, t)$ -secret sharing scheme can be converted into a hierarchical secret sharing protocol (see Section 5.3 Remark 5.3.10).
- (5) A secret cannot be changed easily without changing the shares, because it is a sum over the free length of the elements in the set  $U$  and hence depends on this set.





## Chapter 7

### Private key cryptosystem with $Aut(F)$ (Protocol 8)

In this chapter we introduce **Protocol 8** as well as three modifications of this cryptographic protocol. **Protocol 8** is a private key cryptosystem, which is based on combinatorial group theory. It uses a finitely generated free group  $F$ , a subgroup  $F_U$  of  $F$  with finite rank, a Nielsen reduced set and automorphisms of  $F$ . It is published in [MR16].

In this cryptographic protocol the ciphertext is a sequence of reduced words in  $X$  where the end of each ciphertext unit is marked and  $X$  is a free generating set for a free group  $F$  of finite rank. A modification is given, in which the ciphertext is now only one reduced word in  $X$  instead of a sequence of words, in this case it is possible that additional information is needed for decryption, thus these is sent with the ciphertext if required. In the second modification a faithful representation from  $F$  into the special linear group  $SL(2, \mathbb{Q})$  is used, such that the ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$ . The third modification utilizes the negative solution of Hilbert's Tenth Problem. Instead of a presentation of the ciphertext as a sequence of matrices in  $SL(2, \mathbb{Q})$  the ciphertext is represented as a sequence of matrices in  $GL(2, R)$ , with  $R := \mathbb{Z}[y_1, y_2, \dots, y_n]$ , the integral polynomial ring in  $n \geq 2$  variables.

For the encryption of the plaintext different automorphisms are used. Each plaintext unit is encrypted with another automorphism, as in a One-Time-Pad (see for instance [MvOV97]). The automorphisms are out of a common set  $\mathcal{F}_{Aut} \subset Aut(F)$  (with  $F$  a free group of finite rank). For decryption Bob needs to know which automorphisms of  $\mathcal{F}_{Aut}$  were used for the encryption procedure by Alice. For this choice of elements from  $\mathcal{F}_{Aut}$  regulations are needed. Therefore, Alice and Bob make use of a linear congruence generator with maximal periodic length. Such a generator is also needed for **Protocol 9** and **Protocol 10**.

Thus, we start this chapter with a short introduction of a linear congruence generator. The description of **Protocol 8** and the modifications are explained next. We give for each cryptographic protocol in this chapter a security analysis and beside this we consider chosen plaintext attacks and chosen ciphertext attacks.

For  $n \in \mathbb{N}$  let  $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z}$  be the ring of integers modulo  $n$ . The corresponding residue class in  $\mathbb{Z}_n$  for an integer  $\beta$  is denoted by  $\bar{\beta}$  (see also [BFKR15]).

**Definition 7.0.1.** [BFKR15]

Let  $n \in \mathbb{N}$  and  $\bar{\beta}, \bar{\gamma} \in \mathbb{Z}_n$ . A bijective mapping  $h : \mathbb{Z}_n \rightarrow \mathbb{Z}_n$  given by  $x \mapsto \bar{\beta}x + \bar{\gamma}$  is called a **linear congruence generator**.

**Theorem 7.0.2.** [BFKR15] (Maximal period length for  $n = 2^m$ ,  $m \in \mathbb{N}$ )

Let  $n \in \mathbb{N}$ , with  $n = 2^m$ ,  $m \geq 1$  and let  $\beta, \gamma \in \mathbb{Z}$ , such that  $h : \mathbb{Z}_n \rightarrow \mathbb{Z}_n$ , with  $x \mapsto \bar{\beta}x + \bar{\gamma}$ , is a linear congruence generator. Further let  $\alpha \in \{0, 1, \dots, n-1\}$  be given and  $x_1 = \bar{\alpha}$ ,  $x_2 = h(x_1)$ ,  $x_3 = h(x_2), \dots$ .

Then the sequence  $x_1, x_2, x_3, \dots$  is periodic with maximal periodic length  $n = 2^m$  if and only if the following holds:

1.  $\beta$  is odd, consequently  $\bar{\beta} \neq \bar{0}$ .
2. If  $m \geq 2$  then  $\beta \equiv 1 \pmod{4}$ .
3.  $\gamma$  is odd, consequently  $\bar{\gamma} \neq \bar{0}$ .

Now, we introduce **Protocol 8**. Before Alice and Bob are able to communicate with each other, they have to make some arrangements.

We speak about public parameters also in private key cryptosystems, because these are parameters which each person, also an eavesdropper, Eve, gets, if she looks at the sent ciphertext. Public parameters are also elements, which Alice and Bob communicate with each other publicly. It is also not a secret which plaintext alphabet is used for the communication.

#### Public Parameters

They first agree on the following public parameters.

1. A free group  $F$  with free generating set  $X = \{x_1, x_2, \dots, x_q\}$  with  $q \geq 2$ .
2. A plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  with  $N \geq 2$ .
3. A subset  $\mathcal{F}_{Aut} := \{f_0, f_1, \dots, f_{2^{128}-1}\} \subset Aut(F)$  of automorphisms of  $F$ . It is  $f_i : F \rightarrow F$  and the  $f_i$ ,  $i = 0, 1, \dots, 2^{128} - 1$ , pairwise different, are generated with the help of 0-1-sequences (of different length) and random numbers as described in Section 4.4. The set  $\mathcal{F}_{Aut}$  is part of the key space.
4. They agree on a linear congruence generator  $h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}}$  with a maximal period length (see Definition 7.0.1 and Theorem 7.0.2).

**Remark 7.0.3.** If the set  $\mathcal{F}_{Aut}$  and the linear congruence generator  $h$  are public Alice and Bob are able to change the automorphisms and the linear congruence generator publicly without a private meeting. The set  $\mathcal{F}_{Aut}$  should be large enough to make a brute force search inefficient.

**Variation 7.0.4.** A variation could be, that Alice and Bob choose the number of elements in the starting set  $\mathcal{F}_{Aut}$  less than  $2^{128}$ , say for example  $2^{10}$ . This starting automorphism set  $\mathcal{F}_{Aut}$  should be chosen privately by Alice and Bob as their set of seeds and should not be made public. Then Alice and Bob can extend publicly the starting set  $\mathcal{F}_{Aut}$  to the set  $\mathcal{F}_{Aut_1}$  of automorphisms, such that  $\mathcal{F}_{Aut_1}$  contains, say for example,  $2^{32}$  automorphisms. The number of all elements in  $\mathcal{F}_{Aut_1}$  should make a brute force attack inefficient. The linear congruence generator stays analogous, just the domain and codomain must be adapted to, say for example,  $\mathbb{Z}_{2^{32}}$ . Because of Theorem 7.0.2 Alice and Bob always get a linear congruence generator with maximal periodic length.

---

## Private Parameters

Now, they agree on their private parameters.

1. Alice and Bob choose a free subgroup  $F_U = \langle U \mid \ \rangle$  of  $F$  with rank  $N$  and the free generating set  $U = \{u_1, u_2, \dots, u_N\}$ , with  $U$  a minimal Nielsen reduced set (with respect to a lexicographical order, see for instance Example 4.2.8) and the elements  $u_i \in U$  are freely reduced words in  $X$ . Such systems  $U$  are easily to construct using Theorem 4.2.13 and Lemma 4.2.15 (see also [CgRR08] and [LS77]). It is  $\mathcal{U}_{Nred}$  the set of all minimal Nielsen reduced sets with  $N$  elements in  $F$ , which is part of the key space.
2. They use a one-to-one correspondence between the Nielsen reduced set  $U$  and the plaintext alphabet  $A$ , it is

$$A \rightarrow U$$

$$a_j \mapsto u_j \quad \text{for } j = 1, \dots, N.$$

3. Alice and Bob agree on an automorphism  $f_{\bar{\alpha}} \in \mathcal{F}_{Aut}$ , hence  $\alpha$  is the common secret starting point  $\alpha \in \{0, 1, \dots, 2^{128} - 1\}$ , with  $x_1 = \bar{\alpha} \in \mathbb{Z}_{2^{128}}$ , for the linear congruence generator. With this  $\alpha$  they are able to generate the sequence  $f_{x_1}, f_{x_2}, \dots, f_{x_z}$  (with  $z$  the number of the plaintext units, which are letters from  $A$ ) of automorphisms of the set  $\mathcal{F}_{Aut}$ , which they need for encryption and decryption, respectively.

**The key space:** The set  $\mathcal{U}_{Nred}$  of all minimal (with respect to a lexicographical order) Nielsen reduced subsets of  $F$  with  $N$  elements. The set  $\mathcal{F}_{Aut}$  of  $2^{128}$  randomly chosen automorphisms of  $F$ .

**Variation 7.0.5.** Alice and Bob could use the set  $\mathcal{F}_{Aut}$  more than once, but not exactly in the same way as before:

- (i) They could change privately the seed  $x_1$  for the linear congruence generator. The way how they run through the set  $\mathcal{F}_{Aut}$  stays the same, just the starting point is different.
- (ii) They could change publicly the bijective mapping  $h$  (the seed  $x_1$  stays the same). Just the starting point stays the same, how they run through the set  $\mathcal{F}_{Aut}$  is different.
- (iii) They could change the bijective mapping  $h$  publicly and the seed  $x_1$  privately. Neither the way how they run through the set  $\mathcal{F}_{Aut}$  stays the same nor the starting point.

## Private Key Cryptosystem

Now, we explain the private key cryptosystem in detail and look carefully at the steps for Alice and Bob.

**Public knowledge:**  $F = \langle X \mid \ \rangle$ ,  $X = \{x_1, x_2, \dots, x_q\}$  with  $q \geq 2$ ; plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  with  $N \geq 2$ ; a set  $\mathcal{F}_{Aut} \subset Aut(F)$ ; a linear congruence generator  $h$ .

### **Encryption and Decryption Procedure:**

1. Alice and Bob agree privately on the private parameters: a set  $U \in \mathcal{U}_{Nred}$  with  $|U| = N$  and an automorphism  $f_{\bar{\alpha}} \in \mathcal{F}_{Aut}$ . They also know the one-to-one correspondence between  $U$  and  $A$ .

2. Alice wants to transmit the message

$$S = s_1 s_2 \cdots s_z, \quad z \geq 1,$$

with  $s_i \in A$  to Bob.

2.1. She generates with the linear congruence generator  $h$  and the knowledge of  $f_{\bar{\alpha}}$  the  $z$  automorphisms  $f_{x_1}, f_{x_2}, \dots, f_{x_z}$ , which she needs for encryption. It is  $x_1 = \bar{\alpha}$ ,  $x_2 = h(x_1)$ ,  $\dots$ ,  $x_z = h(x_{z-1})$ .

2.2. The encryption is as follows

$$\text{if } s_i = a_t \quad \text{then } s_i \mapsto c_i := f_{x_i}(u_t), \quad 1 \leq i \leq z, \quad 1 \leq t \leq N.$$

Recall that the one-to-one correspondence  $A \rightarrow U$  with  $a_j \mapsto u_j$ , for  $j = 1, 2, \dots, N$ , holds. The ciphertext

$$\begin{aligned} C &= f_{x_1}(s_1) f_{x_2}(s_2) \cdots f_{x_z}(s_z) \quad \text{with } s_i \hat{=} u_t \Leftrightarrow s_i = a_t \\ &= c_1 c_2 \cdots c_z \end{aligned}$$

is sent to Bob. We call  $c_j$  the ciphertext units, which are words in  $X$ . We do not perform cancellations between  $c_i$  and  $c_{i+1}$  and the end of each  $c_i$  is marked,  $1 \leq i \leq z - 1$ , for example with the symbol “?”.

3. Bob gets the ciphertext

$$C = c_1 \wr c_2 \wr \cdots \wr c_z.$$

He knows where each ciphertext unit  $c_j$  begins and ends. Hence, he gets the information that he has to use  $z$  automorphisms of the set  $\mathcal{F}_{\text{Aut}}$  for decryption. He has now two possibilities for decryption.

3.1.a. With the knowledge of  $f_{\bar{\alpha}}$ , the linear congruence generator  $h$  and the number  $z$ , he computes for each automorphism  $f_{x_i}$ ,  $i = 1, 2, \dots, z$ , the inverse automorphism  $f_{x_i}^{-1}$ .

3.1.b. With the knowledge of  $f_{\bar{\alpha}}$ , the set  $U = \{u_1, u_2, \dots, u_N\}$ , the linear congruence generator  $h$  and the number  $z$ , he computes for each automorphism  $f_{x_i}$ ,  $i = 1, 2, \dots, z$ , the set

$$U_{f_{x_i}} = \{f_{x_i}(u_1), f_{x_i}(u_2), \dots, f_{x_i}(u_N)\}.$$

Hence, with the one-to-one correspondence between  $U$  and  $A$ , he gets a one-to-one correspondence between the letters in the alphabet  $A$  and the words of the ciphertext depending on the automorphisms  $f_{x_i}$ . This is shown in Table 7.1 (page 157). Bob does a search in the table for decryption.

3.2. With the knowledge of the Table 7.1 (page 157) or the inverse automorphisms  $f_{x_i}^{-1}$ , respectively, the decryption is as follows

$$\text{if } c_i = f_{x_i}(u_t) \quad \text{then } c_i \mapsto s_i := f_{x_i}^{-1}(c_i) = a_t, \quad 1 \leq i \leq z, \quad 1 \leq t \leq N.$$

He generates the plaintext message

$$\begin{aligned} S &= f_{x_1}^{-1}(c_1) f_{x_2}^{-1}(c_2) \cdots f_{x_z}^{-1}(c_z) \\ &= s_1 s_2 \cdots s_z \quad \text{with } s_i = a_j \Leftrightarrow s_i \hat{=} u_j, \end{aligned}$$

from Alice.

Table 7.1.: Table for decryption: Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  corresponding to ciphertext alphabet  $U_{f_{x_i}}$  depending on the automorphisms  $f_{x_i}$

	$U_{f_{x_1}}$	$U_{f_{x_2}}$	$\dots$	$U_{f_{x_z}}$
$a_1$	$f_{x_1}(u_1)$	$f_{x_2}(u_1)$	$\dots$	$f_{x_z}(u_1)$
$a_2$	$f_{x_1}(u_2)$	$f_{x_2}(u_2)$	$\dots$	$f_{x_z}(u_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_N$	$f_{x_1}(u_N)$	$f_{x_2}(u_N)$	$\dots$	$f_{x_z}(u_N)$

**Remark 7.0.6.** The cryptosystem is a polyalphabetic system, that means, a word  $u_i \in U$ , and hence a letter  $a_i \in A$ , is encrypted differently at different positions in the plaintext, because different automorphisms are used during the encryption procedure for each ciphertext unit. Thus, for the ciphertext, a statistical frequency attack (see for instance [BFKR15]) over the frequency of words, which corresponds to letters in the plaintext alphabet, or groups of words, is useless.

We summarize **Protocol 8** in Table 7.2 (page 158).

Table 7.2.: Summary of **Protocol 8**: Private key cryptosystem with  $Aut(F)$

<b>Public Knowledge</b>	
$F = \langle X \mid \quad \rangle$ , $X = \{x_1, x_2, \dots, x_q\}$ , $q \geq 2$ ; plaintext alphabet $A = \{a_1, a_2, \dots, a_N\}$ , $N \geq 2$ ; set $\mathcal{F}_{Aut} \subset Aut(F)$ ; linear congruence generator $h$ of maximal periodic length.	
Alice	Bob
Private keys	
Nielsen reduced set $U \subset F$ , $ U  = N$ ; seed $f_{\bar{\alpha}} \in \mathcal{F}_{Aut}$ , one-to-one correspondence $A \rightarrow U$ , $a_j \mapsto u_j$ .	
Encryption	
Choose message $S = s_1 s_2 \cdots s_z, \quad z \geq 1,$ with $s_i \in A$ . Calculate $x_1 = \bar{\alpha}, x_2 = h(x_1), \dots, x_z = h(x_{z-1})$ , obtain $f_{x_1}, f_{x_2}, \dots, f_{x_z}$ . Encryption procedure: if $s_i = a_t$ then $s_i \mapsto c_i := f_{x_i}(u_t)$ , $1 \leq i \leq z$ , $1 \leq t \leq N$ . Ciphertext: $C = f_{x_1}(s_1) f_{x_2}(s_2) \cdots f_{x_z}(s_z) = c_1 c_2 \cdots c_z$ .	$\xrightarrow{C=c_1 c_2 \cdots c_z}$
Decryption	
	Compute $z$ automorphisms: $x_1 = \bar{\alpha}, x_2 = h(x_1), \dots, x_z = h(x_{z-1})$ , obtain $f_{x_1}, f_{x_2}, \dots, f_{x_z}$ . <u>Two possibilities:</u> 1. For each $f_{x_i}$ , $i = 1, 2, \dots, z$ , compute the inverse automorphism $f_{x_i}^{-1}$ . 2. For each $f_{x_i}$ , $i = 1, 2, \dots, z$ , compute $U_{f_{x_i}} = \{f_{x_i}(u_1), f_{x_i}(u_2), \dots, f_{x_i}(u_N)\}$ and get a table like Table 7.1 (page 157). (Decryption: Search in this table.)  With knowledge of Table 7.1 (page 157) or inverse automorphisms $f_{x_i}^{-1}$ , respectively, the decryption is as follows: if $c_i = f_{x_i}(u_t)$ then $c_i \mapsto s_i := f_{x_i}^{-1}(c_i) = u_t$ , $1 \leq i \leq z$ , $1 \leq t \leq N$ . Plaintext message $S = f_{x_1}^{-1}(c_1) f_{x_2}^{-1}(c_2) \cdots f_{x_z}^{-1}(c_z)$ $= s_1 s_2 \cdots s_z, \text{ with } s_i \in A.$

Bob has two possibilities to decrypt the ciphertext. Firstly, he calculates the inverse automorphisms of the  $z$  automorphisms  $f_{x_i}$ , which Alice used for encryption. An example is attached

in Appendix C.7. Secondly, he uses a table (see Table 7.1 (page 157)) for decryption, which stored the ciphertext alphabet  $U_{f_{x_i}}$ , which is generated with the automorphisms Alice used for encryption. The following example performs this last method for decryption.

**Example 7.0.7.** This example was executed with the help of the computer program GAP and the FGA package, see Appendix C.5.

First Alice and Bob agree on the **public parameters**:

1. Let  $F$  be the finitely generated free group on the free generating set  $X = \{a, b, c, d\}$ .
2. Let  $\tilde{A} := \{a_1, a_2, \dots, a_{12}\} = \{A, E, I, O, U, T, M, L, K, Y, B, S\}$  be the plaintext alphabet.
3. A set  $\mathcal{F}_{Aut}$  is determined. In this example we give the automorphisms, which Alice and Bob use for encryption and decryption, respectively, just at the moment when they are needed.
4. The linear congruence generator with maximal periodic length is

$$h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}} \\ x \mapsto \bar{5}x + \bar{3}.$$

The **private parameters** for this example are:

1. The free subgroup  $F_{\tilde{U}}$  of  $F$  with the free generating set

$$\tilde{U} = \{u_1, u_2, \dots, u_{12}\} \\ = \{ba^2, cd, d^2c^{-2}, a^{-1}b, a^4b^{-1}, b^3a^{-2}, bc^3, bc^{-1}bab^{-1}, c^2ba, c^2dab^{-1}, a^{-1}d^3c^{-1}, a^2db^2d^{-1}\}.$$

The set  $\tilde{U}$  is a Nielsen reduced set and the group  $F_{\tilde{U}}$  has rank 12. Alice and Bob agree on the starting automorphism  $f_{\bar{93}}$ , hence it is  $x_1 = \bar{\alpha} = \bar{93}$ . It is known, that  $a_i \mapsto u_i$ ,  $i = 1, 2, \dots, 12$ , for  $u_i \in \tilde{U}$  and  $a_i \in \tilde{A}$ , therefore

$$\begin{array}{llll} A \hat{=} u_1 = ba^2, & E \hat{=} u_2 = cd, & I \hat{=} u_3 = d^2c^{-2}, & O \hat{=} u_4 = a^{-1}b, \\ U \hat{=} u_5 = a^4b^{-1}, & T \hat{=} u_6 = b^3a^{-2}, & M \hat{=} u_7 = bc^3, & L \hat{=} u_8 = bc^{-1}bab^{-1}, \\ K \hat{=} u_9 = c^2ba, & Y \hat{=} u_{10} = c^2dab^{-1}, & B \hat{=} u_{11} = a^{-1}d^3c^{-1}, & S \hat{=} u_{12} = a^2db^2d^{-1}. \end{array}$$

We look at the **encryption and decryption procedure** for Alice and Bob.

2. With the above agreements **Alice** is able to encrypt her message

$$S = \text{ILIKEBOB.}$$

Her message is of length 8. She generates the ciphertext as follows:

- 2.1. She first determines, with the help of the linear congruence generator  $h$ , the automorphisms  $f_{x_i}$ ,  $i = 1, 2, \dots, 8$ , which she needs for encryption. It is

$$\begin{array}{lll} x_1 = \bar{\alpha} = \bar{93}, & x_2 = h(x_1) = \bar{468}, & x_3 = h(x_2) = \bar{2343}, \\ x_4 = h(x_3) = \bar{11718}, & x_5 = h(x_4) = \bar{58593}, & x_6 = h(x_5) = \bar{292968}, \\ x_7 = h(x_6) = \bar{1464843}, & x_8 = h(x_7) = \bar{7324218}. \end{array}$$

The automorphisms are describable with the help of regular Nielsen transformations, it is

$$f_{x_1} \doteq (N1)_3(N2)_{1.4}(N2)_{4.3}(N2)_{2.3}(N1)_3(N2)_{1.4}(N2)_{3.1},$$

$$\begin{aligned} f_{x_1} : F &\rightarrow F \\ a &\mapsto ad^2c^{-1}, \quad b \mapsto bc^{-1}, \quad c \mapsto cad^2c^{-1}, \quad d \mapsto dc^{-1}; \end{aligned}$$

$$f_{x_2} \doteq (N2)_{1.4}(N1)_2(N2)_{2.4}(N2)_{3.1}(N1)_2(N1)_1(N2)_{1.3}[(N2)_{4.3}]^2(N1)_3,$$

$$\begin{aligned} f_{x_2} : F &\rightarrow F \\ a &\mapsto d^{-1}a^{-1}cad, \quad b \mapsto d^{-1}b, \quad c \mapsto d^{-1}a^{-1}c^{-1}, \quad d \mapsto d(cad)^2; \end{aligned}$$

$$f_{x_3} \doteq (N1)_2(N2)_{4.2}(N1)_4(N2)_{2.4}(N1)_2(N2)_{4.2}(N1)_3(N2)_{2.1}(N2)_{3.2} \\ [(N2)_{1.4}]^3(N1)_2(N2)_{4.2},$$

$$\begin{aligned} f_{x_3} : F &\rightarrow F \\ a &\mapsto ab^3, \quad b \mapsto a^{-1}d^{-1}, \quad c \mapsto c^{-1}da, \quad d \mapsto ba^{-1}d^{-1}; \end{aligned}$$

$$f_{x_4} \doteq [(N2)_{3.1}]^2(N1)_2[(N2)_{2.1}]^3(N2)_{2.4}(N2)_{4.2}(N2)_{1.3},$$

$$\begin{aligned} f_{x_4} : F &\rightarrow F \\ a &\mapsto aca^2, \quad b \mapsto b^{-1}a^3d, \quad c \mapsto ca^2, \quad d \mapsto db^{-1}a^3d; \end{aligned}$$

$$f_{x_5} \doteq (N2)_{1.2}(N1)_3(N1)_1[(N2)_{4.3}]^2(N2)_{1.2}(N1)_2(N1)_3(N2)_{2.4}(N2)_{3.1},$$

$$\begin{aligned} f_{x_5} : F &\rightarrow F \\ a &\mapsto b^{-1}a^{-1}b, \quad b \mapsto b^{-1}dc^{-2}, \quad c \mapsto cb^{-1}a^{-1}b, \quad d \mapsto dc^{-2}; \end{aligned}$$

$$f_{x_6} \doteq (N1)_1(N2)_{2.3}(N2)_{3.1}(N1)_2(N2)_{1.2}(N2)_{4.2},$$

$$\begin{aligned} f_{x_6} : F &\rightarrow F \\ a &\mapsto a^{-1}c^{-1}b^{-1}, \quad b \mapsto c^{-1}b^{-1}, \quad c \mapsto ca^{-1}, \quad d \mapsto dc^{-1}b^{-1}; \end{aligned}$$

$$f_{x_7} \doteq [(N2)_{2.1}]^3(N1)_3[(N2)_{4.3}]^3(N1)_1(N2)_{1.2}(N1)_2(N2)_{2.4}(N2)_{3.1},$$

$$\begin{aligned} f_{x_7} : F &\rightarrow F \\ a &\mapsto a^{-1}ba^3, \quad b \mapsto a^{-3}b^{-1}dc^{-3}, \quad c \mapsto c^{-1}a^{-1}ba^3, \quad d \mapsto dc^{-3}; \end{aligned}$$

$$f_{x_8} \doteq (N2)_{1.4}(N1)_2(N1)_3(N2)_{2.1}[(N2)_{3.4}]^2(N1)_4(N1)_1(N1)_3(N2)_{4.2},$$

$$\begin{aligned} f_{x_8} : F &\rightarrow F \\ a &\mapsto d^{-1}a^{-1}, \quad b \mapsto b^{-1}ad, \quad c \mapsto d^{-2}c, \quad d \mapsto d^{-1}b^{-1}ad. \end{aligned}$$

Note, that the regular Nielsen transformations are applied from the left to the right.



2.2 The ciphertext is now

$$\begin{aligned}
C &= f_{x_1}(I)f_{x_2}(L)f_{x_3}(I)f_{x_4}(K)f_{x_5}(E)f_{x_6}(B)f_{x_7}(O)f_{x_8}(B) \\
&= f_{x_1}(d^2c^{-2})f_{x_2}(bc^{-1}bab^{-1})f_{x_3}(d^2c^{-2})f_{x_4}(c^2ba)f_{x_5}(cd)f_{x_6}(a^{-1}d^3c^{-1}) \\
&\quad f_{x_7}(a^{-1}b)f_{x_8}(a^{-1}d^3c^{-1}) \\
&= dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1} \wr d^{-1}bcabd^{-1}a^{-1}caddb^{-1}d \wr \\
&\quad (ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2 \wr (ca^2)^2b^{-1}a^3daca^2 \wr cb^{-1}a^{-1}bdc^{-2} \wr \\
&\quad bca(dc^{-1}b^{-1})^3ac^{-1} \wr a^{-1}(a^{-2}b^{-1})^2dc^{-3} \wr (ab^{-1})^3adc^{-1}d^2 \\
&= c_1c_2c_3c_4c_5c_6c_7c_8.
\end{aligned}$$

The symbol “ $\wr$ ” marks the end of each ciphertext unit  $c_i$ ,  $1 \leq i \leq z - 1$ .

3. **Bob** gets the ciphertext

$$\begin{aligned}
C &= dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1} \wr d^{-1}bcabd^{-1}a^{-1}caddb^{-1}d \wr \\
&\quad (ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2 \wr (ca^2)^2b^{-1}a^3daca^2 \wr cb^{-1}a^{-1}bdc^{-2} \wr \\
&\quad bca(dc^{-1}b^{-1})^3ac^{-1} \wr a^{-1}(a^{-2}b^{-1})^2dc^{-3} \wr (ab^{-1})^3adc^{-1}d^2
\end{aligned}$$

from Alice. Now, he knows, that he needs eight automorphisms for decryption.

3.1. Bob knows the set  $U$ , the linear congruence generator  $h$  and the starting seed automorphism  $f_{\overline{93}}$ . For decryption he uses tables (analogous to Table 7.1 (page 157)).

Now, he is able to compute for each automorphism  $f_{x_i}$  the set  $U_{f_{x_i}}$ ,  $i = 1, 2, \dots, 8$ , and to generate the tables Table 7.3 (page 161), Table 7.4 (page 162), Table 7.5 (page 162) and Table 7.6 (page 163).

Table 7.3.: Correspondence: plaintext alphabet to ciphertext alphabet I

	$U_{f_{x_1}}$	$U_{f_{x_2}}$
A	$b(c^{-1}ad^2)^2c^{-1}$	$d^{-1}bd^{-1}a^{-1}c^2ad$
E	$cad(dc^{-1})^2$	$d^{-1}a^{-1}c^{-1}(dca)^2d$
I	$dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1}$	$((dca)^2d)^2cadcad$
O	$cd^{-2}a^{-1}bc^{-1}$	$d^{-1}a^{-1}c^{-1}ab$
U	$(ad^2c^{-1})^3ad^2b^{-1}$	$d^{-1}a^{-1}c^4adb^{-1}d$
T	$(bc^{-1})^2bd^{-2}a^{-1}cd^{-2}a^{-1}$	$(d^{-1}b)^3d^{-1}a^{-1}c^{-2}ad$
M	$b(ad^2)^3c^{-1}$	$d^{-1}b(d^{-1}a^{-1}c^{-1})^3$
L	$bd^{-2}a^{-1}c^{-1}bc^{-1}ad^2b^{-1}$	$d^{-1}bcabd^{-1}a^{-1}caddb^{-1}d$
K	$c(ad^2)^2c^{-1}bc^{-1}ad^2c^{-1}$	$(d^{-1}a^{-1}c^{-1})^2d^{-1}bd^{-1}a^{-1}cad$
Y	$c(ad^2)^2c^{-1}dc^{-1}ad^2b^{-1}$	$(d^{-1}a^{-1}c^{-1})^2dcadc^2adb^{-1}d$
B	$cd^{-2}a^{-1}(dc^{-1})^2d^{-1}a^{-1}c^{-1}$	$d^{-1}a^{-1}c^{-1}(ad^2cadc)^3adcad$
S	$(ad^2c^{-1})^2d(c^{-1}b)^2d^{-1}$	$d^{-1}a^{-1}c^2ad(dca)^2bd^{-1}b(d^{-1}a^{-1}c^{-1})^2d^{-1}$

Table 7.4.: Correspondence: plaintext alphabet to ciphertext alphabet II

	$U_{f_{x_3}}$	$U_{f_{x_4}}$
A	$a^{-1}d^{-1}(ab^3)^2$	$b^{-1}a^3d(aca^2)^2$
E	$c^{-1}daba^{-1}d^{-1}$	$ca^2db^{-1}a^3d$
I	$(ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2$	$(db^{-1}a^3d)^2a^{-2}c^{-1}a^{-2}c^{-1}$
O	$b^{-3}a^{-2}d^{-1}$	$a^{-2}c^{-1}a^{-1}b^{-1}a^3d$
U	$(ab^3)^4da$	$(aca^2)^4d^{-1}a^{-3}b$
T	$(a^{-1}d^{-1})^3(b^{-3}a^{-1})^2$	$(b^{-1}a^3d)^3a^{-2}c^{-1}a^{-3}c^{-1}a^{-1}$
M	$a^{-1}d^{-1}(c^{-1}da)^3$	$b^{-1}a^3d(ca^2)^3$
L	$(a^{-1}d^{-1})^2ca^{-1}d^{-1}ab^3da$	$b^{-1}a^3da^{-2}c^{-1}b^{-1}a^3daca^2d^{-1}a^{-3}b$
K	$c^{-1}dac^{-1}ab^3$	$(ca^2)^2b^{-1}a^3daca^2$
Y	$(c^{-1}da)^2ba^{-1}d^{-1}ab^3da$	$(ca^2)^2db^{-1}a^3daca^2d^{-1}a^{-3}b$
B	$b^{-3}a^{-1}(ba^{-1}d^{-1})^3a^{-1}d^{-1}c$	$a^{-2}c^{-1}a^{-1}(db^{-1}a^3d)^3a^{-2}c^{-1}$
S	$(ab^3)^2b(a^{-1}d^{-1})^2b^{-1}$	$aca^3c(a^2db^{-1}a)^2a^2$

Table 7.5.: Correspondence: plaintext alphabet to ciphertext alphabet III

	$U_{f_{x_5}}$	$U_{f_{x_6}}$
A	$b^{-1}dc^{-2}b^{-1}a^{-2}b$	$(c^{-1}b^{-1}a^{-1})^2c^{-1}b^{-1}$
E	$cb^{-1}a^{-1}bdc^{-2}$	$ca^{-1}dc^{-1}b^{-1}$
I	$dc^{-2}dc^{-1}(c^{-1}b^{-1}ab)^2c^{-1}$	$(dc^{-1}b^{-1})^2ac^{-1}ac^{-1}$
O	$b^{-1}adc^{-2}$	$bcac^{-1}b^{-1}$
U	$b^{-1}a^{-4}bc^2d^{-1}b$	$(a^{-1}c^{-1}b^{-1})^3a^{-1}$
T	$(b^{-1}dc^{-2})^3b^{-1}a^2b$	$(c^{-1}b^{-1})^2abca$
M	$b^{-1}dc^{-1}(b^{-1}a^{-1}bc)^2b^{-1}a^{-1}b$	$c^{-1}b^{-1}(ca^{-1})^3$
L	$b^{-1}dc^{-2}b^{-1}abc^{-1}b^{-1}dc^{-2}b^{-1}a^{-1}bc^2d^{-1}b$	$c^{-1}b^{-1}ac^{-2}b^{-1}a^{-1}$
K	$cb^{-1}a^{-1}bcb^{-1}a^{-1}dc^{-2}b^{-1}a^{-1}b$	$ca^{-1}c(a^{-1}c^{-1}b^{-1})^2$
Y	$(cb^{-1}a^{-1}b)^2dc^{-2}b^{-1}a^{-1}bc^2d^{-1}b$	$(ca^{-1})^2dc^{-1}b^{-1}a^{-1}$
B	$b^{-1}ab(dc^{-2})^3b^{-1}abc^{-1}$	$bca(dc^{-1}b^{-1})^3ac^{-1}$
S	$b^{-1}a^{-2}b(dc^{-2}b^{-1})^2$	$(a^{-1}c^{-1}b^{-1})^2d(c^{-1}b^{-1})^2d^{-1}$

Table 7.6.: Correspondence: plaintext alphabet to ciphertext alphabet IV

	$U_{f_{x_7}}$	$U_{f_{x_8}}$
A	$a^{-3}b^{-1}dc^{-3}a^{-1}(ba^2)^2a$	$b^{-1}d^{-1}a^{-1}$
E	$c^{-1}a^{-1}ba^3dc^{-3}$	$d^{-2}cd^{-1}b^{-1}ad$
I	$dc^{-3}dc^{-3}(a^{-3}b^{-1}ac)^2$	$d^{-1}(b^{-1}a)^2(dc^{-1}d)^2d$
O	$a^{-1}(a^{-2}b^{-1})^2dc^{-3}$	$adb^{-1}ad$
U	$a^{-1}(ba^2)^4ac^3d^{-1}ba^3$	$(d^{-1}a^{-1})^5b$
T	$(a^{-3}b^{-1}dc^{-3})^3a^{-1}(a^{-2}b^{-1})^2a$	$(b^{-1}ad)^3adad$
M	$a^{-3}b^{-1}dc^{-3}(c^{-1}a^{-1}ba^3)^3$	$b^{-1}a(d^{-1}cd^{-1})^2d^{-1}c$
L	$a^{-3}b^{-1}dc^{-3}a^{-3}b^{-1}aca^{-3}b^{-1}dc^{-3}a^{-1}ba^3c^3d^{-1}ba^3$	$b^{-1}adc^{-1}d^2b^{-1}d^{-1}a^{-1}b$
K	$c^{-1}a^{-1}ba^3c^{-1}a^{-1}dc^{-3}a^{-1}ba^3$	$(d^{-2}c)^2b^{-1}$
Y	$(c^{-1}a^{-1}ba^3)^2dc^{-3}a^{-1}ba^3c^3d^{-1}ba^3$	$(d^{-2}c)^2d^{-1}b^{-1}d^{-1}a^{-1}b$
B	$a^{-3}b^{-1}a(dc^{-3})^3a^{-3}b^{-1}ac$	$(ab^{-1})^3adc^{-1}d^2$
S	$a^{-1}(ba^2)^2a(dc^{-3}a^{-3}b^{-1})^2$	$(d^{-1}a^{-1})^2d^{-1}(b^{-1}ad)^2d$

3.2. With these tables he is able to generate the plaintext from Alice, it is

$$\begin{aligned}
 S &= f_{x_1}^{-1} (dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1}) f_{x_2}^{-1} (d^{-1}bcabd^{-1}a^{-1}cadb^{-1}d) \\
 &\quad f_{x_3}^{-1} ((ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2) f_{x_4}^{-1} ((ca^2)^2b^{-1}a^3daca^2) \\
 &\quad f_{x_5}^{-1} (cb^{-1}a^{-1}bdc^{-2}) f_{x_6}^{-1} (bca(dc^{-1}b^{-1})^3ac^{-1}) \\
 &\quad f_{x_7}^{-1} (a^{-1}(a^{-2}b^{-1})^2dc^{-3}) f_{x_8}^{-1} ((ab^{-1})^3adc^{-1}d^2) \\
 &\hat{=} \text{ILIKEBOB.}
 \end{aligned}$$

**Security 7.0.8.** An eavesdropper, Eve, intercepts the ciphertext

$$C = c_1c_2 \cdots c_z,$$

with  $c_i = f_{x_i}(u_j)$  for some  $1 \leq j \leq N$ . This is a **ciphertext only attack** also called known ciphertext attack (see Section 1.1 or for instance [BFKR15], [MvOV97] or [BNS10]). If Alice and Bob choose non characteristic subgroups, then it is likely that  $c_j \notin F_U$  for some  $1 \leq j \leq z$ . In general, the sets  $U_{f_{x_i}}$  are not Nielsen equivalent sets to  $U$  under the automorphisms  $f_{x_i}$ . Hence, the ciphertext units give no hint for the subgroup  $F_U$ . Eve knows  $L = \sum_{k=1}^z |c_k|$ , the length of  $C$ , because Alice does not perform cancellations between  $c_i$  and  $c_{i+1}$ , for  $1 \leq i \leq z - 1$ . To break the system Eve needs to know the set  $U$ . For this it is likely that she assumes that the ball  $B(F, L)$  in the Cayley graph for  $F$  contains a basis for  $F_U$ . With this assumption she searches for primitive elements  $y$  for  $F_U$  in the ball  $B(F, L)$ ,  $|y| \leq L$ ,  $y \in F$ . In fact she needs to find  $N$  primitive elements for  $F_U$  in  $B(F, L)$  (these would be primitive elements for  $F_U$  in a ball  $B(F_U, L)$  for some Nielsen reduced basis for  $F_U$ ). From Proposition 4.2.21 and Theorem 4.2.22 it is known that the number of primitive elements grows exponentially with the free length of the elements. Eve chooses sets  $V_i := \{v_{i_1}, v_{i_2}, \dots, v_{i_K}\}$  with  $K \geq N$  and elements  $v_{i_j}$  in  $B(F, L)$  and with Nielsen transformations she constructs the corresponding Nielsen reduced sets

$V'_i$  (which are minimal concerning a lexicographical order). If  $|V'_i| = N$  then  $V'_i$  is a candidate for  $U$ .

The running time is within  $\mathcal{O}(\lambda^2 K^2)$ , with  $\lambda := \max\{|v_{j_\ell}|_X \mid v_{j_\ell} \in V_j \text{ for } \ell = 1, 2, \dots, K\} \leq L$ , to get a Nielsen reduced set  $V'_j$  from  $V_j$  with the algorithm in [Ste89] (see Remark 4.2.17).

How can Eve verify that her generated set  $V'_i =: U'$  (with  $U' = \{u'_1, u'_2, \dots, u'_N\}$ ), which is a candidate for  $U$ , is the used set  $U$  by Alice and Bob?

Therefor she calculates

$$U'_{f_i} = \{f_i(u'_1), f_i(u'_2), \dots, f_i(u'_N)\},$$

$1 \leq i \leq |\mathcal{F}_{\text{Aut}}|$ , and proves for each  $i$  if there is a  $j \in \{1, 2, \dots, N\}$ , such that  $c_\ell = f_i(u'_j)$ , if she finds such an  $i$  and  $j$  Eve assumes that she found an automorphism in  $\mathcal{F}_{\text{Aut}}$ , which Alice used in her encryption, Eve is now able to prove if  $c_{\ell+1} = f_{h(i)}(u'_{k'})$  for a  $k' \in \{1, 2, \dots, N\}$ , ( $h$  is the public linear congruence generator) therefore she calculates

$$U'_{f_{h(i)}} = \{f_{h(i)}(u'_1), f_{h(i)}(u'_2), \dots, f_{h(i)}(u'_N)\}.$$

If  $\ell > 1$  Eve is also able to prove if  $c_{\ell-1} = f_{h^{-1}(i)}(u'_{k'})$  for a  $k' \in \{1, 2, \dots, N\}$ . The mapping  $h$  is a public linear congruence generator of maximal periodic length, hence bijective and therefore the inverse mapping of  $h$  exists. She calculates

$$U'_{f_{h^{-1}(i)}} = \{f_{h^{-1}(i)}(u'_1), f_{h^{-1}(i)}(u'_2), \dots, f_{h^{-1}(i)}(u'_N)\}.$$

In general if for all  $c_k$ ,  $1 \leq k \leq z$ , follows that  $c_{\ell+j} = f_{h^j(i)}(u'_{k'})$  for some  $k' \in \{1, 2, \dots, N\}$ , with  $h^j(i) = \underbrace{h(h(\dots h(h(i))\dots))}_{j \text{ times } h}$  if  $j > 0$  and  $h^j(i) = \underbrace{h^{-1}(h^{-1}(\dots h^{-1}(h^{-1}(i))\dots))}_{|j| \text{ times } h^{-1}}$  if  $j < 0$ ,

respectively, then it is very likely that  $U = U'$  and Eve is able to read the message because she knows (can calculate with the linear congruence generator) the automorphisms which were used for encryption and she knows (or is able to calculate with the help of a frequency attack)  $a_i \mapsto u_i$ ,  $1 \leq i \leq N$ , and the plaintext alphabet  $A$  is public. This is a brute force search through the public set  $\mathcal{F}_{\text{Aut}}$ .

Another idea for Eve is to calculate the inverse automorphisms of the known set  $\mathcal{F}_{\text{Aut}}$  and apply these inverse automorphisms  $f_j^{-1}$ ,  $j = 0, 1, \dots, 2^{128} - 1$ , to the ciphertext units  $c_i$ ,  $i = 1, \dots, z$ . If she fixes  $c_k$  and calculates  $f_j^{-1}(c_k)$  for all  $j = 0, 1, \dots, 2^{128} - 1$  then at least one of these elements is a correct element in  $U$ , namely  $f_{x_k}^{-1}(c_k)$ , but she does not know which  $j \in \{0, 1, \dots, 2^{128} - 1\}$  is  $x_k$ . To a given ciphertext

$$C = c_1 c_2 \dots c_z,$$

with  $c_i = f_{x_i}(u_j)$  for some  $1 \leq j \leq N$ , she could calculate the possible plaintext

$$S_\ell^{\text{invers}} = f_\ell^{-1}(c_1) f_{h(\ell)}^{-1}(c_2) f_{h^2(\ell)}^{-1}(c_3) \dots f_{h^{z-1}(\ell)}^{-1}(c_z),$$

with  $\ell \in \{0, 1, \dots, 2^{128}\}$  and the public linear congruence generator  $h$ . Thus, Eve gets  $2^{128}$  possible plaintexts written as elements in  $F$ . If the eavesdropper gets more than  $N$  different words in the plaintext  $S_\ell^{\text{invers}}$  she knows, that this is not a correct candidate for the plaintext. The plaintext consist of maximum  $N$  different letters, because the plaintext alphabet consist of  $N$  letters. In each situation where the ciphertext is rewritten in maximum  $N$  different words in  $F$  Eve could do a statistical frequency attack (see for instance [BFKR15]) over the frequency of letters (or groups of letters). Then she is able to generate possible plaintexts, whereby the

correct plaintext is contained. To prevent this kinds of attack it is possible to keep parts of the set or the whole set  $\mathcal{F}_{Aut}$  private, such that Eve does not know from which automorphism she should calculate the inverse elements.

**Remark 7.0.9.** Alice assumes that the ball  $B(F, L)$  in the Cayley graph for  $F$  contains a basis for  $F_U$  or at least all elements in  $U$  which were used for the encryption. She cannot be sure that this is true because the automorphisms for encryption are automorphisms on  $F$  and hence therefore it is likely that  $c_i \notin F_U$ . It is possible that there exists an element  $u_i \in U$ , which was used for the encryption, and it is  $|u_i|_X \geq L$ , with  $L = \sum_{k=1}^z |c_k|$ , thus Eve is not able to find the element  $u_i$  in the ball  $B(F, L)$  of the Cayley graph from  $F$ . For a small example see Appendix B.3.

**Remark 7.0.10.** Eve uses  $L = \sum_{k=1}^z |c_k|$  for her search in the Cayley graph (see Security 7.0.8) because it is likely, that the elements of  $U$  lay in the ball  $B(F, L)$ . Another number which she could use is  $L_1 = \max\{|c_k| \mid 1 \leq k \leq z\}$ . This number  $L_1$  would make the search for primitive elements in the Cayley graph faster, because

$$\max\{|c_k| \mid 1 \leq k \leq z\} \leq L = \sum_{i=1}^z |c_k|$$

and the number of primitive elements in the ball  $B(F, L)$  grows exponentially with the free length of the elements (see Proposition 4.2.21 and Theorem 4.2.22). Hence, it is faster to search in the ball  $B(F, L_1)$  if

$$\max\{|u_j| \mid 1 \leq j \leq N\} \leq \max\{|c_k| \mid 1 \leq k \leq z\}. \quad (7.1)$$

Eve cannot be sure that inequality (7.1) is true for the ciphertext units and the elements in the Nielsen reduced set  $U$ , because there exists automorphisms, such that the inequality does not hold (see Appendix B.1) and Eve cannot decide if she is in such a situation.

The main security certification depends on the fact, that for a single subset of  $K \geq N$  elements Eve finds a Nielsen reduced set in the running time  $\mathcal{O}(\lambda^2 K^2)$  but she has to test all possible subsets of  $K$  elements for which she needs exponential running time, because the number of primitive elements grows exponentially with the free length, see Proposition 4.2.21 and Theorem 4.2.22. She searches in a ball  $B(F, L)$ , with  $L = \sum_{i=1}^z |c_i|$  for these primitive elements.

To verify the set  $U'$  as  $U$  or to find the automorphisms for  $s_i$ , which were used by Alice and Bob, (and hence decrypt the message) Eve could do a brute force search through the set  $\mathcal{F}_{Aut}$ .

**Remark 7.0.11.** If the set  $\mathcal{F}_{Aut}$  is private Eve has no hints for the used automorphisms of  $Aut(F)$ , which were used by Alice. Even if Eve gets a candidate  $U'$  for  $U$  she does not know which automorphism is used and she also has no set  $\mathcal{F}_{Aut}$  which she could use for a brute force search. In general, the elements  $c_i$  are no elements of  $F_U$  hence she cannot use the algorithm (in Theorem 4.3.10), which solves the constructive membership problem to get a hint for the automorphism.

Alice and Bob made the set  $\mathcal{F}_{Aut}$  public, because they are then able to change the automorphisms without a private meeting, see Remark 7.0.3. If they use Variation 7.0.4 they have a public part of the set  $\mathcal{F}_{Aut}$  and a private part, hence they make an attack for Eve more difficult than in the situation when the set  $\mathcal{F}_{Aut}$  is completely public.

**Remark 7.0.12.** The one-to-one correspondence between  $A$  and  $U$  is not public. If Eve is able to decrypt the ciphertext to a version where she can write it as a kind of plaintext with elements in  $U$ , then she can use a statistical frequency attack (see for instance [BFKR15]) to reconstruct

the correspondence between  $A$  and  $U$  and hence read the plaintext. Thus, it is possible that Alice and Bob let the theoretical one-to-one correspondence between  $A$  and  $U$  remain public. Note that they do not publish the explicit set  $U$ .

**Remark 7.0.13.** If Alice and Bob used an arbitrary Nielsen reduced set  $U$  and not a minimal Nielsen reduced set corresponding to a lexicographical order, then Eve gets for  $V'_i$  much more sets, see Example 4.2.16. She gets also sets  $V'_i$  which are the set  $U$  but with permuted order. In this case she has to test all permuted sets and not only the minimal set (corresponding to a lexicographical order), thus she gets  $N!-1$  more sets for each  $V'_i$  to test with the automorphisms. There are also Nielsen reduced sets, which generate the same group but differ not only in the permutation order but also in some elements, for example  $\{y^2, y^{-1}xy\}$  and  $\{y^2, yxy^{-1}\}$  generate the same free group (see Example 4.2.16).

The security certification can be improved with the next three modifications, which are explained in Section 7.1, Section 7.2 and Section 7.3.

## 7.1. Modification with the ciphertext a reduced word for the cryptosystem with $Aut(F)$

We present a modification where the ciphertext is only one reduced word in  $X$  instead of a sequence of words, in this case it is possible that additional information is needed for decryption, thus these is sent with the ciphertext if required.

Let  $C$ , with

$$C = c_1 c_2 \cdots c_z,$$

be the unreduced ciphertext to the plaintext

$$S = s_1 s_2 \cdots s_z,$$

with  $s_i \in A$  and  $A$  the used plaintext alphabet, as described above in **Protocol 8**, a private key cryptosystem with  $Aut(F)$ . Let  $F = \langle X \mid \quad \rangle$  be the finitely generated free group for Alice and Bob, with  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ , and let  $U = \{u_1, u_2, \dots, u_N\}$  be the used Nielsen reduced set with  $u_i$  words in  $X$  and  $N = |A|$ . Remember, that there is a one-to-one correspondence between  $A$  and  $U$ , it is  $a_i \mapsto u_i$ , for  $1 \leq i \leq N$ . It is possible that  $c_i$  ends with  $x_k$  and  $c_{i+1}$  begins with  $x_k^{-1}$ , with  $x_k \in X^{\pm 1}$ , thus in  $C$  there is a part with  $x_k x_k^{-1}$ , for a  $1 \leq k \leq q$ . The end of each ciphertext unit is marked, such that Bob and also the eavesdropper, Eve, know where each  $c_j$  ends and begins and they also know the whole reduced word  $c_j$ .

In this modification the reduced word  $C_{red}$  of the ciphertext  $C$  is sent instead of  $C$ , hence there are no parts with  $x_k x_k^{-1}$ , for a  $1 \leq k \leq q$ , which only occur in  $C$  if  $x_k$  is the last letter of  $c_j$  and  $x_k^{-1}$  is the first letter of  $c_{j+1}$ . The beginning and end of each ciphertext unit  $c_i$  is not marked. Let

$$C_{red} = x'_1 x'_2 \cdots x'_{L'},$$

with  $x'_i \in X^{\pm 1}$ ,  $1 \leq i \leq L'$ , be the reduced ciphertext from Alice for Bob.

Now, if Bob wants to decrypt the ciphertext  $C_{red}$  he first calculates

$$U_{f_{x_1}} = \{f_{x_1}(u_1), f_{x_1}(u_2), \dots, f_{x_1}(u_N)\}.$$

His aim is to find the first ciphertext unit  $c_1$  in  $C_{red}$ . There could occur different cases for him. Depending on the occurred cases Alice may have to give additional information to Bob. This additional information is of the form

$$(i, d, \tilde{w})$$

with the following impact

$i$  :  $i$  states the ciphertext unit  $c_i = \tilde{x}_1 \tilde{x}_2 \cdots \tilde{x}_{p_i}$ ,  $\tilde{x}_j \in X^{\pm 1}$ , to which the additional information corresponds;

$d$  :  $d$  states the letter  $\tilde{x}_d$  in  $c_i$  after which Bob then knows that the word  $\tilde{w}$  appears in  $c_i$ ;

$\tilde{w}$  :  $\tilde{w}$  is the word which appears in  $c_i$  after  $\tilde{x}_d$ , such that the identification of  $c_i$  is uniquely for Bob.

It is  $i \in \{1, 2, \dots, z\}$ ,  $d \in \{0, 1, 2, \dots, p_i\}$  and  $\tilde{w}$  a reduced word in  $X$  or the empty word.

This information is specified and explained in the following different cases, which could occur for a ciphertext  $C_{red}$ .

(i) It is  $C_{red} \equiv c_1 w$  with  $w$  a word in  $X$ . That means,  $c_1$  is an initial segment of  $C_{red}$  and no letters of  $c_1$  are canceled. There are two subcases:

a) Assume there are  $1 \leq j, t \leq N$ , with  $j \neq t$ , such that

$$f_{x_1}(u_j) \equiv f_{x_1}(u_t) x' \omega_2,$$

with  $x' \in X^{\pm 1}$  and  $\omega_2$  a word in  $X$  or the empty word, which is 1. The ciphertext  $C_{red}$  is of the form

$$\begin{aligned} C_{red} &\equiv c_1 w \\ &\equiv f_{x_1}(u_j) \omega_1 \\ &\equiv f_{x_1}(u_t) x' \omega_2 \omega_1, \end{aligned}$$

with  $\omega_1$  a word in  $X$ . If  $w = \omega_1$  then  $c_1 = f_{x_1}(u_j)$ ; if  $w \equiv x' \omega_2 \omega_1$  then  $c_1 = f_{x_1}(u_t)$ . Bob does not know if Alice encrypted with  $c_1$  the plaintext letter  $a_j$  or  $a_t$  (remember the one-to-one correspondence with  $a_i \mapsto u_i$ ) because it could be, that in  $U_{f_{x_2}}$  exists an element  $f_{x_2}(u_k) \equiv x' \omega_2 \omega_3$ , with  $\omega_3$  an initial segment of  $\omega_1$  (or  $\omega_3 = \omega_1$  if the plaintext consists of two plaintext units, that means  $z = 2$ ).

If  $c_1 = f_{x_1}(u_t)$  Alice gives Bob the information

$$(1, |f_{x_1}(u_t)|, 1).$$

If  $c_1 = f_{x_1}(u_j)$  Alice gives Bob the information

$$(1, |f_{x_1}(u_t)|, x') \quad \text{or} \quad (1, |f_{x_1}(u_j)|, 1).$$

With this information Bob is able to identify  $c_1$  uniquely.

If there occurs a case in which Bob gets this problem with more than two elements in  $U_{f_{x_1}}$  the situation is similar. Just the element  $x'$  is then not a letter but a word in  $X$ , which makes a uniquely identification of the correct element in  $U_{f_{x_1}}$  possible for Bob, see Appendix B.2.

b) Assume  $c_1 = f_{x_1}(u_j)$  and it is

$$f_{x_1}(u_t) \neq f_{x_1}(u_j)x'\omega_2 \quad \text{and} \quad f_{x_1}(u_j) \neq f_{x_1}(u_t)x'\omega_2,$$

for all  $j \neq t$  with  $1 \leq j \leq N$ . Bob needs no additional information from Alice. He can uniquely identify  $c_1$ .

(ii) There are cancellations between  $c_1$  and  $c_2$ . Thus we get two subcases:

a) The ciphertext unit  $c_1$  is completely canceled by  $c_2$ :

It is  $c_2 \equiv c_1^{-1}w$  with  $w$  a word in  $X$  or the empty word. Therefore, Bob is not able to find the ciphertext unit  $c_1$  in the initial segment of  $C_{red}$ . In this case Alice gives Bob the additional information

$$(1, 0, \tilde{c}_1),$$

with  $\tilde{c}_1$  an initial segment of  $c_1$ , such that  $c_1$  is uniquely identifiable in the set  $U_{f_{x_1}}$ . In the worst case it is  $\tilde{c}_1 = c_1$  which only occurs if  $U_{f_{x_1}}$  has at least two elements  $f_{x_1}(u_j)$  and  $f_{x_1}(u_t)$  as in (i) a) as well as  $c_1 = f_{x_1}(u_t)$  or if  $c_1 = f_{x_1}(u_j)$  and  $|f_{x_1}(u_j)| = |f_{x_1}(u_t)| + 1$ .

b) The ciphertext unit  $c_1$  is not completely canceled by  $c_2$ :

Let  $c_1 \equiv \omega_1 x' \omega_2$  and  $c_2 \equiv \omega_2^{-1} x'^{-1} \omega_3$ , with  $\omega_1, \omega_2, \omega_3$  words in  $X$ ,  $x' \in X^{\pm 1}$  and no cancellations between  $\omega_1$  and  $\omega_3$ ; ( $\omega_1 \neq 1$ ). Alice has to give additional information if the ciphertext unit  $c_1$  is not uniquely identifiable for Bob in the set  $U_{f_{x_1}}$  with the knowledge of the initial segment  $\omega_1$  of  $c_1$ . This is the case if the set  $U_{f_{x_1}}$  has at least two elements  $f_{x_1}(u_k)$  and  $f_{x_1}(u_\ell)$  with  $f_{x_1}(u_k) \equiv \omega_1 w$  and  $f_{x_1}(u_\ell) \equiv \omega_1 w'$ , with  $w$  and  $w'$  words in  $X$ . Then Alice gives as additional information

$$(1, |\omega_1|, x' \tilde{\omega}_2),$$

with  $\tilde{\omega}_2$  an initial segment of  $\omega_2$  which is long enough to identify  $c_1$  uniquely in  $U_{f_{x_1}}$ . Or Alice gives the information

$$(1, |c_1|, 1)$$

if  $c_1$  is the only element in  $U_{f_{x_1}}$  of free length  $|c_1|$ . For an example see Example B.2.1 in Appendix B.2.

After this Bob works with  $C_{red}^{(2)} = c_1^{-1} C_{red}$  and  $U_{f_{x_2}}$  to identify  $c_2$ ; the above decrypted cases (i) and (ii) occur analogously. After this he works with  $C_{red}^{(3)} = c_2^{-1} C_{red}^{(2)}$  and  $U_{f_{x_3}}$  and so on until he found all ciphertext units  $c_i$  and hence decrypted the ciphertext  $C_{red}$ . The cases (i) and (ii) appear analogously for  $C_{red}^{(2)}, C_{red}^{(3)}, \dots, C_{red}^{(z-1)}$ . It is  $C_{red}^{(z)} = c_{z-1}^{-1} C_{red}^{(z-1)} = c_z$  which can be found in  $U_{f_{x_z}}$  uniquely.

**Remark 7.1.1.** The number  $d$  in the additional information  $(i, d, \tilde{w})$ , with  $\tilde{w} \neq 1$ , tells Bob that the first  $d$  letters of  $c_i$  are the first  $d$  letters of  $C_{red}^{(i)}$ .

**Remark 7.1.2.** If Alice gives the information  $(i, |c_i|, 1)$  Bob knows that the element  $c_i$  is an initial segment of  $C_{red}^{(i)}$ , if there are more elements in  $U_{f_{x_i}}$  of free length  $|c_i|$ . Or if  $c_i$  is the only element in  $U_{f_{x_i}}$  of free length  $|c_i|$  then Bob can uniquely identify  $c_i$  also if a terminal segment of  $c_i$  is canceled by an initial segment of  $c_{i+1}$  in  $C_{red}^{(i)}$  and hence in  $C_{red}$ .

**Remark 7.1.3.** Additional information from Alice for Bob is only required

- if a ciphertext unit  $c_i$  is completely canceled by the ciphertext unit  $c_{i+1}$ ;



- if  $c_i = f_{x_i}(u_k)$  and there is at least another element  $f_{x_i}(u_\ell)$  in the set  $U_{f_{x_i}}$  with either  $c_i = f_{x_i}(u_\ell)\omega_1$  or  $c_i\omega_2 = f_{x_i}(u_\ell)$  with  $\omega_1$  and  $\omega_2$  words in  $X$ . An example of such an automorphism is given in Appendix B.2;
- if there are cancellations between  $c_i = \omega_1 x' \omega_2$  and  $c_{i+1} = \omega_2^{-1} x'^{-1} \omega_3$ , with  $\omega_1, \omega_2, \omega_3$  words in  $X$ ,  $x' \in X^{\pm 1}$ , and no cancellations between  $\omega_1$  and  $\omega_3$ , and the set  $U_{f_{x_1}}$  has at least two elements  $f_{x_i}(u_k)$  and  $f_{x_i}(u_\ell)$  with  $f_{x_i}(u_k) = \omega_1 w$  and  $f_{x_i}(u_\ell) = \omega_1 w'$ , with  $w$  and  $w'$  words in  $X$ , see Example B.2.1 in Appendix B.2

**Example 7.1.4.** In Example 7.0.7 the ciphertext is

$$\begin{aligned} C = & c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8 \\ & dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1} \wr d^{-1}bcabd^{-1}a^{-1}caddb^{-1}d \wr \\ & (ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2 \wr (ca^2)^2b^{-1}a^3daca^2 \wr cb^{-1}a^{-1}bdc^{-2} \wr \\ & bca(dc^{-1}b^{-1})^3ac^{-1} \wr a^{-1}(a^{-2}b^{-1})^2dc^{-3} \wr (ab^{-1})^3adc^{-1}d^2 \end{aligned}$$

and the reduced ciphertext is

$$\begin{aligned} C_{red} = & dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1}d^{-1}bcabd^{-1}a^{-1}caddb^{-1}d(ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2(ca^2)^2b^{-1} \\ & a^3daca^2cb^{-1}a^{-1}bdc^{-2}bca(dc^{-1}b^{-1})^3ac^{-1}a^{-1}(a^{-2}b^{-1})^2dc^{-3}(ab^{-1})^3adc^{-1}d^2. \end{aligned}$$

In this example are no reductions between the ciphertext units  $c_i$  and all ciphertext units are uniquely identifiable in the corresponding set  $U_{f_{x_i}}$ . Thus, no additional information is needed from Alice for decryption. In the first moment Bob does not know how many sets  $U_{f_{x_i}}$  he will need and hence how big the table (a table like Table 7.1 (page 157)) will be, which he needs for decryption, but he knows the set  $U$ , the starting automorphism  $f_{\bar{a}}$ , the used set  $\mathcal{F}_{Aut}$  as well as the linear congruence generator  $h$ , hence he is able to calculate the required sets  $U_{f_{x_i}}$  for the table. These tables are given in Example 7.0.7, see Table 7.3 (page 161), Table 7.4 (page 162), Table 7.5 (page 162) and Table 7.6 (page 163).

Note, that the decryption is done with tables like Table 7.1 (page 157). If Bob wants to use the inverse automorphisms  $f_{x_i}^{-1}$  of the used automorphisms  $f_{x_i} \in \mathcal{F}_{Aut}$  of Alice, he has to know where each ciphertext unit  $c_i$  ends and begins and he also needs to know the canceled letters in the word  $C_{red}$ . Thus, the additional information which Bob needs should give him exactly the unreduced ciphertext  $C$  with the markers “ $\wr$ ” between the ciphertext units. Therefore, also Eve gets this information and is able to reconstruct the unreduced ciphertext  $C$ , hence this modification has then no advantage over the unreduced version. Thus, Bob has to use the tables for encryption. Even if there are no cancellations between the ciphertext units and only the markers “ $\wr$ ” are missing, Bob does not know on how many letters in  $C_{red}$  even the first inverse automorphism is to apply. It could be that  $f_{x_1}^{-1}(c_1) = u_3$  and  $f_{x_1}^{-1}(c_2) = u_3^{-1}u_5$  and Bob applies  $f_{x_1}^{-1}$  on  $c_1$  and  $c_2$  and thus, gets for example

$$f_{x_1}^{-1}(c_1 c_2) = f_{x_1}^{-1}(c_1) f_{x_1}^{-1}(c_2) = (u_3)(u_3^{-1}u_5) = u_5,$$

so  $s_1 \hat{=} u_5$  instead of  $s_1 \hat{=} u_3$  (which would be correct). It could also be, that  $c_1 \equiv w_{11}w_{12}$  and  $c_2 \equiv w_{21}w_{22}$ , with  $w_{11}, w_{12}, w_{21}$  and  $w_{22}$  words in  $X$  and  $f_{x_1}^{-1}(w_{11}) = u_2$  and Bob stops and gets for  $f_{x_2}^{-1}(w_{12}w_{21}w_{22}) = f_{x_2}^{-1}(w_{12})f_{x_2}^{-1}(w_{21}w_{22}) = (u_6u_3^{-1})u_3 = u_6$ , but correct would be

$$f_{x_1}^{-1}(w_{11}w_{12}) = f_{x_1}^{-1}(w_{11})f_{x_1}^{-1}(w_{12}) = (u_2)(u_2^{-1}u_4) = u_4$$

and

$$f_{x_2}^{-1}(w_{2_1} w_{2_2}) = u_3.$$

In the wrong decryption Bob would get  $S = a_2 a_6 \tilde{a}$  with  $\tilde{a}$  a word in  $A$  but correct would be  $S = a_4 a_3 \tilde{a}'$ , with  $\tilde{a}'$  a word in  $A$ .

**Remark 7.1.5.** We take a look at the situation for the eavesdropper, Eve, and her information which she obtains from the additional information which Alice sends to Bob. Eve knows from

1.  $(i, |c_i|, 1)$ , that there is at least one element in  $U_{f_{x_i}}$  of free length  $|c_i|$ , but she does not know if this element is completely visibly in  $C_{red}$  or  $C_{red}^{(i)}$ ;
2.  $(i, |c'_i|, w)$ , with  $c'_i$  an initial segment of  $c_i$  and  $w$  an segment of  $c_i$  ( $c_i \equiv c'_i w \tilde{w}$  with  $\tilde{w}$  word in  $X$  or the empty word), that  $|c_i| \geq |c'_i| + |w|$  (it is  $|c_i| = |c'_i| + |w|$  if  $\tilde{w} = 1$ ), and the first  $|c'_i|$  elements of  $C_{red}^{(i)}$  are the first  $|c'_i|$  elements of  $c_i$  and after these elements comes the word  $w$  in  $c_i$ , but she does not know, if this word  $w$  is visible in  $C_{red}$  or  $C_{red}^{(i)}$ ;
3.  $(i, 0, c'_i)$ , with  $c'_i$  the initial segment of  $c_i$ , that the word  $c_i$  is completely canceled in  $C_{red}$ . She also knows that  $c'_i$  is an initial segment of  $c_i$ , thus  $|c_i| \geq |c'_i|$ , it is not necessary that  $c'_i = c_i$ .

In general Eve cannot be sure where  $C_{red}^{(i)}$  begins (this is equivalent to the beginning of  $c_i$ ) or where  $c_i$  ends. She is also not able to identify all missing letters of  $X^{\pm 1}$  in  $C_{red}$ , which she needs to get the unreduced word  $C$ .

**Security 7.1.6.** An eavesdropper, Eve, intercepts the reduced ciphertext

$$C_{red} = x'_1 x'_2 \cdots x'_L,$$

with  $x'_i \in X^{\pm 1}$ ,  $1 \leq i \leq L'$ . In general, she is not able to identify the end of  $c_1$  and hence she cannot identify the beginning or end of the other ciphertext units  $c_j$ ,  $2 \leq j \leq z$ , and she also does not know which elements are canceled in the reduced ciphertext  $C_{red}$ .

As in Security 7.0.8 to break the system an eavesdropper, Eve, needs to know the set  $U$ . She knows  $L' = |C_{red}|_X$  the freely reduced length of the reduced ciphertext. For this it is likely that she assumes, as above, that the ball  $B(F, L')$  in the Cayley graph for  $F$  contains a basis for  $F_U$ . She searches in the same way for candidates of  $U$  as explained in Security 7.0.8.

Now it is difficult for Eve to verify that she gets the correct candidate  $U' = \{u'_1, u'_2, \dots, u'_N\}$  for  $U$ , because she does not know where each ciphertext unit  $c_j$  ends or begins. Furthermore, she does not know if there are cancellations between the ciphertext units and hence if there are letters in the ciphertext  $C_{red}$  missing from which she has no idea. Maybe the additional information from Alice, which is sent publicly to Bob, gives her hints, but she cannot be sure that there are no other letters missing, which she cannot deduce from the additional information, see Remark 7.1.5. Hence, if she goes the way described in Security 7.0.8, then it is not likely that, even if she gets an automorphism  $f_i \in \mathcal{F}_{Aut}$  and found in

$$U'_{f_i} = \{f_i(u'_1), f_i(u'_2), \dots, f_i(u'_N)\}$$

a  $j$ , such that  $f_i(u'_j)$  is a segment of  $C_{red}$ , this automorphism  $f_i$  is a used automorphism for encryption and it is not clear that she is on the right way even if in

$$U'_{f_{h(i)}} = \{f_{h(i)}(u'_1), f_{h(i)}(u'_2), \dots, f_{h(i)}(u'_N)\}$$

the next segment of  $C_{red}$  can be found.

Eve is not able to identify the ciphertext units  $c_i$  uniquely and completely. Bob is able to do it uniquely and completely because he is able to calculate the sets  $U_{f_{x_i}}$  and hence tables, like Table 7.1 (page 157), with which he is able to decrypt the ciphertext.

## 7.2. Modification with $\text{SL}(2, \mathbb{Q})$ for the cryptosystem with $\text{Aut}(F)$

In this modification Alice and Bob use the fact, that there is no algorithm known to solve the membership problem (see Problem 4.3.8) for (discrete) free subgroups of  $\text{SL}(2, \mathbb{Q})$  which are of rank greater than or equal to 2 and not subgroups of  $\text{SL}(2, \mathbb{Z})$ , Remark 4.3.14 (see [EKL14]). Let the initial set up be exactly as described in the beginning of Chapter 7. Thus,  $F$  is a finitely generated free group on the free generating set  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ , the set  $A = \{a_1, a_2, \dots, a_N\}$ ,  $N \geq 2$ , is the used alphabet for the plaintext and  $U = \{u_1, u_2, \dots, u_N\}$  is a Nielsen reduced set with  $u_j$ ,  $1 \leq j \leq N$ , words in  $X$ .

Alice and Bob agree in addition to **Protocol 8** privately on a faithful representation of a finitely generated free group  $F$  into  $\text{SL}(2, \mathbb{Q})$ , that is,  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ , thus they can write the ciphertext units  $c_i$  of the ciphertext  $C$  as matrices in  $\text{SL}(2, \mathbb{Q})$  instead as words in  $X$ . Let

$$\begin{aligned} \varphi : F &\rightarrow \text{SL}(2, \mathbb{Q}) \\ x_i &\mapsto M_i, \end{aligned}$$

be this faithful representation of  $F$  into  $\text{SL}(2, \mathbb{Q})$  (see Remark 7.2.2 for a proposal to generate  $M_i$ ). The group  $F_\varphi = \langle M_1, M_2, \dots, M_q \mid \ \rangle$  is isomorphic to  $F$  under the mapping  $x_i \mapsto M_i$ , for  $i = 1, \dots, q$ . Alice and Bob take care that the free matrix group  $F_\varphi$  is not a subgroup of  $\text{SL}(2, \mathbb{Z})$ . The ciphertext is now

$$\begin{aligned} C' &= \varphi(c_1)\varphi(c_2)\cdots\varphi(c_z) \\ &= W_1W_2\cdots W_z, \end{aligned}$$

a sequence of matrices  $W_j \in \text{SL}(2, \mathbb{Q})$ . The matrices  $W_j$  are words in

$$X_\varphi := \{\varphi(x_1), \varphi(x_2), \dots, \varphi(x_q)\} = \{M_1, M_2, \dots, M_q\}.$$

The encryption is realizable with a table (like Table 7.1 (page 157)) if the representation  $\varphi$  is applied to the elements  $f_{x_i}(u_j)$  in the table. In general, it is not possible to use the inverse automorphism of  $f_{x_i}$  for decryption because Bob does not know how the matrix  $W_i = \varphi(c_i)$  is written as a product of the matrices  $M_1, M_2, \dots, M_q$  (constructive membership problem is not solvable) but this knowledge is important to apply the inverse automorphism  $f_{x_i}^{-1}$  on the matrix  $W_i = \varphi(c_i) = \varphi(f_{x_i}(u_j))$  and then to reconstruct  $\varphi(u_j)$  and hence  $u_j$  (and then  $s_i = a_j$ ). Therefore, Bob gets the Table 7.7 (page 172) with matrices and hence an assignment from the matrices to the plaintext alphabet depending on the automorphisms  $f_{x_i}$  and the faithful representation  $\varphi$ , with

$$U_{\varphi(f_{x_i})} = \{\varphi(f_{x_i}(u_1)), \varphi(f_{x_i}(u_2)), \dots, \varphi(f_{x_i}(u_N))\} \subset \text{SL}(2, \mathbb{Q}).$$

Table 7.7.: Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  corresponding to ciphertext alphabet  $U_{\varphi(f_{x_i})}$  depending on the automorphisms  $f_{x_i}$  and the faithful representation  $\varphi$

	$U_{\varphi(f_{x_1})}$	$U_{\varphi(f_{x_2})}$	$\dots$	$U_{\varphi(f_{x_z})}$
$a_1$	$\varphi(f_{x_1}(u_1))$	$\varphi(f_{x_2}(u_1))$	$\dots$	$\varphi(f_{x_z}(u_1))$
$a_2$	$\varphi(f_{x_1}(u_2))$	$\varphi(f_{x_2}(u_2))$	$\dots$	$\varphi(f_{x_z}(u_2))$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_N$	$\varphi(f_{x_1}(u_N))$	$\varphi(f_{x_2}(u_N))$	$\dots$	$\varphi(f_{x_z}(u_N))$

**Security 7.2.1.** Eve intercepts a ciphertext, which is a sequence of matrices

$$C' = W_1 W_2 \dots W_z,$$

with  $W_i \in \text{SL}(2, \mathbb{Q})$ ,  $1 \leq i \leq z$ . To get a situation as in Security 7.0.8 and hence to be able to start an attack, Eve has to write each matrix  $W_i$ ,  $1 \leq i \leq z$ , as a word in  $M$ , which is the set  $M = \{M_1, M_2, \dots, M_q\}$ , with  $M_i = \varphi(x_i)$ , because  $M_i$  corresponds to  $x_i$ .

Eve makes a guess  $\tilde{M} = \{\tilde{M}_1, \tilde{M}_2, \dots, \tilde{M}_q\}$  for the set  $M$ , that means a guess for the faithful representation  $\varphi$ . To be sure, that  $\tilde{M}$  is a qualified candidate for  $M$  she has to solve the membership problem for all ciphertext matrices  $W_i$ ,  $1 \leq i \leq z$ , and the set  $\tilde{M}$  (see Problem 4.3.8). If she is also able to solve the constructive membership problem for all ciphertext matrices  $W_i$ ,  $1 \leq i \leq z$ , and the set  $\tilde{M}$ , she is then in a situation as explained in Security 7.0.8, that is, she can write  $C'$  as an unreduced word in  $X$ . If  $\tilde{M} = M$  she gets  $C$  otherwise she gets another unreduced word in  $X$  but not necessary  $C$ . If Alice and Bob take care, that no algorithm is known to solve the membership problem for  $W_i$ ,  $1 \leq i \leq z$ , and  $\{M_1, M_2, \dots, M_q\}$  then it is very unlikely that Eve is able to decrypt the message correctly.

If there is no algorithm known to solve the membership problem for the group  $F_M = \langle M \mid \_ \rangle$ , Eve could only do a brute force search, that means, she makes a guess  $\tilde{M} = \{\tilde{M}_1, \tilde{M}_2, \dots, \tilde{M}_q\}$  for the set  $M$  (which is a guess for the faithful representation  $\varphi$  which is then  $\varphi' : F \rightarrow \text{SL}(2, \mathbb{Q})$ ) and a guess  $U'$  for the set  $U$ , which is Nielsen reduced of cardinality  $N$  and has elements, which are words in  $X$ . She has no hints for these sets. She then could use the automorphisms  $f_i \in \mathcal{F}_{\text{Aut}}$  to calculate with her sets  $U'$  and  $\tilde{M}$  (and hence with  $\varphi'$ ) sets  $U'_{\varphi'(f_i)}$ . She then searches for matrices  $W_j$ ,  $1 \leq j \leq z$ , which are ciphertext matrices of  $C'$  in her generated sets  $U'_{\varphi'(f_i)}$ .

Even if she gets for her set  $U'$  and  $\tilde{M}$  the matrix  $W_j = \varphi'(f_i(u'_k))$  she cannot be sure that this is correct for the plaintext from Alice, that means  $W_i$  is decrypted to  $a_k$ . There are a lot of sets  $\tilde{M} \neq M$ , which Eve could choose for  $M$ , and also a lot of sets of  $U' \neq U$  which she could use for  $U$ , such that  $W_j = \varphi'(f_i(u'_k))$ , for some  $j, i$  and  $k$ , with  $1 \leq j \leq z$ ,  $1 \leq i \leq 2^{128}$  and  $1 \leq k \leq N$ . It is difficult for Eve to recognize if she gets the correct sets  $U$  and  $M$ , if there is no algorithm known to solve the membership problem for  $F_M = \langle M \mid \_ \rangle$ .

Hence, here the additional security certification is, that there is no algorithm known to solve the membership problem (see Problem 4.3.8) for (discrete) free subgroups of  $\text{SL}(2, \mathbb{Q})$ , which are of rank greater than or equal to 2 and not subgroups of  $\text{SL}(2, \mathbb{Z})$  (see Remark 4.3.14 and the

paper [EKL14]).

**Remark 7.2.2.** If Alice and Bob use Theorem 4.2.18 to generate the matrices they first should take care that  $q \geq 3$ , because Theorem 4.2.18 generates a representation of a discontinuous group (see [Leh64, pp. 246]) and discontinuous groups are discrete [Leh64, Theorem on p. 96]). B. Eick, M. Kirschmer and C. Leedham-Green present in the paper [EKL14] a practical algorithm to solve the constructive membership problem (see Problem 4.3.9) for discrete free subgroups of rank 2 of  $SL(2, \mathbb{R})$  (see Remark 4.3.14). Second, they should take care, that they do not use  $q$  matrices for the set  $M$  which are generated by Theorem 4.2.18. They should calculate a set  $X' = \{X_1, X_2, \dots, X_\ell\}$ ,  $\ell \geq 3$ , with matrices generated by Theorem 4.2.18, and the set  $M$  should be a free generating set (not necessary Nielsen reduced) with  $q$  elements for a subgroup of  $\langle X' \mid \ \rangle$ . It is  $M = \{M_1, M_2, \dots, M_q\}$ , with  $M_j$  a word in  $X'$  and  $|M_j|_{X'} \geq 2$ , for all  $j = 1, 2, \dots, q$ . Alice and Bob should also take care, that  $\langle M \mid \ \rangle$  is not a subgroup of  $SL(2, \mathbb{Z})$ , because the membership problem is effectively solvable in  $SL(2, \mathbb{Z})$ , see Theorem 4.3.13. The matrices in  $X'$  have all a special look, which is

$$X_j = \begin{pmatrix} -r_j & -1 + r_j^2 \\ 1 & -r_j \end{pmatrix},$$

with  $r_j \in \mathbb{Q}$ ,  $1 \leq j \leq q$ , and it is

$$r_{j+1} - r_j \geq 3 \quad \text{and} \quad r_1 \geq 2.$$

Maybe Eve could start an attack with this information if  $X' = M$ .

The encryption and decryption is realized with Table 7.7 (page 172), which is dynamically expandable, such that it is possible for Alice and Bob to generate a table for each message they want to communicate. After calculating this table, the encryption and decryption procedure is just a search in this table. We now mention a way to modify the ciphertext.

**Variation 7.2.3.** Instead of the whole matrix  $\varphi(f_{x_i}(u_j)) = W_i = \begin{pmatrix} w_{i1} & w_{i2} \\ w_{i3} & w_{i4} \end{pmatrix}$ ,  $1 \leq i \leq z$  and  $1 \leq j \leq N$ , they use one entry  $w_{i_k}$  of the matrix as ciphertext unit  $c_i$ . They use for the ciphertext unit  $c_i$ ,  $1 \leq i \leq z$ , the element

$$\begin{aligned} w_{i_1} & \text{ if } i \equiv 1 \pmod{4}, \\ w_{i_2} & \text{ if } i \equiv 2 \pmod{4}, \\ w_{i_3} & \text{ if } i \equiv 3 \pmod{4}, \\ w_{i_4} & \text{ if } i \equiv 0 \pmod{4}. \end{aligned}$$

Assume that  $i \equiv e \pmod{4}$  and there are two matrices  $W_i$  and  $\tilde{W}_i$  in  $U_{\varphi(f_{x_i})}$  with entries  $w_{i_e} = \tilde{w}_{i_e}$  then it is not clear which matrix and hence which letter in the alphabet  $A$  is encrypted with this element. In this situation the encrypter gives in addition the next entry of the correct matrix  $W_i$  until the matrix  $W_i$  is uniquely identifiable. At least if all four entries of a matrix are given, it is uniquely identifiable and there is no other matrix in  $U_{\varphi(f_{x_i})}$  with exactly these four entries because  $U_{\varphi(f_{x_i})}$  is a basis.

If the Variation 7.2.3 is used, they send instead of matrices just rational numbers and hence they need less space for the ciphertext than in the case if they send the whole matrices. With Table 7.7 (page 172) it is possible to decrypt the ciphertext. If Alice and Bob proof that the matrices  $\varphi(f_{x_i}(u_j)) = N_j = \begin{pmatrix} n_{j1} & n_{j2} \\ n_{j3} & n_{j4} \end{pmatrix}$  in the column of  $U_{\varphi(f_{x_i})}$  have at the place  $y \in \{1, 2, 3, 0\}$ , with  $i \equiv y \pmod{4}$ , all different entries  $n_{j_y}$  then they only need to store these entries instead of

the whole matrix and hence need less space to store the table. In addition, if an eavesdropper intercepts the ciphertext she gets for  $c_i$  no element of the set  $U_{f_{x_i}}$ . She just gets an entry of one matrix of this set and this makes an attack nearly impossible.

**Example 7.2.4.** In this example (see Appendix C.6 for calculations in Maple 16 and GAP) Alice and Bob agree additionally to Example 7.0.7 on a faithful representation. With Theorem 4.2.18 they generate the matrices

$$X_1 := \begin{pmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{pmatrix}, \quad X_2 := \begin{pmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{pmatrix} \quad \text{and} \quad X_3 := \begin{pmatrix} \frac{-23}{2} & \frac{525}{4} \\ 1 & \frac{-23}{2} \end{pmatrix}.$$

These matrices form a basis for a free group  $F_{\varphi_1}$  of rank 3. Alice and Bob generate a subgroup  $F_\varphi$  of  $F_{\varphi_1}$  of rank 4 because  $X = \{a, b, c, d\}$ . The free generating set for  $F_\varphi$  is the set  $M = \{X_1X_2, X_3X_1^2, X_2X_3X_2, X_1^{-1}X_2\}$ . Thus, the faithful representation is

$$\begin{aligned} \varphi : F &\rightarrow \text{SL}(2, \mathbb{Q}) \\ a &\mapsto X_1X_2 = \begin{pmatrix} \frac{75}{2} & \frac{-1111}{4} \\ -11 & \frac{163}{2} \end{pmatrix}, & b &\mapsto X_3X_1^2 = \begin{pmatrix} -1189 & 3990 \\ 104 & -349 \end{pmatrix}, \\ c &\mapsto X_2X_3X_2 = \begin{pmatrix} -2681 & 19966 \\ 360 & -2681 \end{pmatrix}, & d &\mapsto X_1^{-1}X_2 = \begin{pmatrix} 15 & -109 \\ 4 & -29 \end{pmatrix}. \end{aligned}$$

The ciphertext is now

$$\begin{aligned} C' &= \varphi(dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1}) \varphi(d^{-1}bcabd^{-1}a^{-1}cadb^{-1}d) \\ &\quad \varphi((ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2) \varphi((ca^2)^2b^{-1}a^3daca^2) \varphi(cb^{-1}a^{-1}bdc^{-2}) \\ &\quad \varphi(bca(dc^{-1}b^{-1})^3ac^{-1}) \varphi(a^{-1}(a^{-2}b^{-1})^2dc^{-3}) \varphi((ab^{-1})^3adc^{-1}d^2) \\ &= \begin{pmatrix} \frac{-429743093559909}{2} & \frac{-6400784021410159}{4} \\ -62588240305379 & \frac{-932216979117085}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{-3240070331754423030683243991}{2} & \frac{47007695458416827592369656315}{4} \\ -223326322203710575272321977 & \frac{3240070327830150751386194361}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{-6899014060703475554169965}{2} & \frac{102756972145191520348785607}{4} \\ 301722468685102729969483 & \frac{-4493988131847945704997109}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{-397074726172421275253684843812134445}{2} & \frac{5883318761059670223751985896578473377}{4} \\ 26659253089426526822952736194350493 & \frac{-395000924306510751052288425218790757}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{46475888407425825}{2} & \frac{692232489736400389}{4} \\ -3120351373297111 & \frac{-46475896943687759}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{-37154085868492177463035768197599}{2} & \frac{-553374013794643763898030444104547}{4} \\ 1624906569753714749910956723073 & \frac{24201404758781402065719318991873}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{-3418963163764785449276501363}{2} & \frac{-50923553357916815212095363641}{4} \\ -230751369629481141540301125 & \frac{-3436913216344813651054341083}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{2739747352948144349387}{2} & \frac{-39628644296581967709615}{4} \\ -402070084312200114547 & \frac{5815679440792026855107}{2} \end{pmatrix}. \end{aligned}$$

Instead of a sequence of words in  $F$  Alice sends to Bob a sequence of eight matrices in  $\text{SL}(2, \mathbb{Q})$ .

The matrices, which are needed for encryption and are therefore stored in a table like Table 7.7 (page 172), are computed in Appendix C.6 in the ‘‘Decryption’’ part.

With Variation 7.2.3 the ciphertext is

$$C'' = \frac{-429743093559909}{2} \wr \frac{47007695458416827592369656315}{4} \wr \frac{301722468685102729969483}{2} \wr \frac{-395000924306510751052288425218790757}{2} \wr \frac{46475888407425825}{2} \wr \frac{-553374013794643763898030444104547}{4} \wr \frac{-230751369629481141540301125}{2} \wr \frac{5815679440792026855107}{2}$$

The symbol “ $\wr$ ” marks the end of a ciphertext unit  $c_i$ .

### 7.3. Modification with Hilbert's Tenth Problem for the cryptosystem with $\text{Aut}(F)$

In this modification the negative solution of Hilbert's Tenth Problem is used.

Hilbert's Tenth Problem: *Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.*

This is the tenth problem of a list of problems (see [Hil02]) which Hilbert presented 1900 at the International Congress of Mathematicians in Paris. In 1970 Hilbert's Tenth Problem was finally proved to be negative by Y. Matiyasevich in [Mat70]. For this he used a series of results by J. Robinson, M. Davis and H. Putnam (see for example [Dav73] or [Mat70] or [Mat96]).

Alice and Bob use instead of a presentation of the ciphertext in  $\text{SL}(2, \mathbb{Q})$  a presentation of the ciphertext in a finitely generated free group in  $\text{GL}(2, R)$ , with  $R := \mathbb{Z}[y_1, y_2, \dots, y_n]$ , the integral polynomial ring in the variables  $y_1, y_2, \dots, y_n$  with  $n \geq 2$ .

This modification is inspired by the public key cryptosystem AMC1, which is explained in [BFKR15, Chapter 12] and [BF08]. We now recall some needed theory of  $\text{PSL}(2, \mathbb{Z})$  and augmented rings.

#### Properties of $\text{PSL}(2, \mathbb{Z})$ :

Firstly, it is known that  $\text{PSL}(2, \mathbb{Z})$ , the group of  $2 \times 2$  projective integral matrices of determinant 1, is finitely presented, that is,

$$\text{PSL}(2, \mathbb{Z}) = \langle s, t \mid s^2 = (st)^3 = 1 \rangle,$$

with

$$s = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \text{and} \quad t = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},$$

see Remark 4.3.12. Secondly, it is known, that the matrices

$$a := t^2 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad b := (st^{-1}s)^2 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$$

freely generate a free subgroup  $J$  of  $\text{PSL}(2, \mathbb{Z})$ , thus  $J = \langle a, b \mid \ \ \rangle$ . Each subgroup of  $J$  is free, see Theorem 4.2.14, moreover  $J$  contains subgroups of every finite rank.

Thirdly, it is known that there exists an algorithms, see Remark 4.3.12, to write each matrix  $T \in \text{PSL}(2, \mathbb{Z})$  in terms of  $s$  and  $t$ .

### Augmented rings:

**Definition 7.3.1.** [BFKR15, Definition 12.4.1]

An **augmented ring** is a unitary ring  $R$  together with a unitary homomorphism

$$\epsilon : R \rightarrow \mathbb{Z}.$$

**Remark 7.3.2.** [BFKR15]

A ring  $R$  with a multiplicative identity 1 becomes an augmented ring if it contains an ideal  $I$ , the augmented ideal, and  $R/I$  is isomorphic to  $\mathbb{Z}$ . It follows that we can view  $\mathbb{Z}$  as a subring of  $R$  and thus

$$R = \mathbb{Z} \oplus I.$$

**Example 7.3.3.** We give three examples for augmented rings

1.  $\mathbb{Z}$ , which is a trivial example;
2.  $\mathbb{Z}[y_1, y_2, \dots, y_n]$ , which is the ring of integral polynomials in any number  $n \in \mathbb{N}$  of variables.
3.  $\mathbb{Z}[G]$ , which is an integral group ring of a group  $G$ .

With the next lemma and its corollaries we can connect such augmented rings (Example 7.3.3 2. and 3.) to free groups and then we can use Hilbert's Tenth Problem for our cryptosystems.

**Lemma 7.3.4.** [BFKR15, Lemma 12.4.2]

Let  $R$  and  $S$  be unitary rings and let  $\phi$  be a homomorphism from  $R$  to  $S$ . If  $\text{GL}(m, R)$  is the group of all  $m \times m$  matrices over  $R$ , then  $\phi$  induces a homomorphism  $\phi^*$  of  $\text{GL}(m, R)$  into  $\text{GL}(m, S)$ .

**Corollary 7.3.5.** [BFKR15, Lemma 12.4.3]

If  $R$  is an augmented ring with augmentation  $\epsilon$ , then the augmentation  $\epsilon$  from  $R$  into  $\mathbb{Z}$  induces a homomorphism  $\epsilon^*$  from  $\text{GL}(n, R)$  to  $\text{GL}(n, \mathbb{Z})$ .

**Corollary 7.3.6.** [BFKR15, Lemma 12.4.3]

Suppose that  $\phi$  is a unitary homomorphism of the unitary ring  $R$  into the unitary ring  $S$  and that  $X$  is a subset of  $\text{GL}(m, R)$ . If  $\phi^*(X)$  freely generates a free subgroup of  $\text{GL}(m, S)$ , then  $X$  freely generates a free subgroup of  $\text{GL}(m, R)$ .

The important consequence of the corollary is

**Lemma 7.3.7.** [BFKR15, Lemma 12.4.5]

Let  $R$  be an augmented ring and

$$r_1, r_2, r_3, r_4, r, r_6, r_7, r_8 \in R.$$

Furthermore, let

$$A = \begin{pmatrix} r_1 & r_2 \\ r_3 & r_4 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} r_5 & r_6 \\ r_7 & r_8 \end{pmatrix}.$$



If

$$\begin{aligned} \epsilon(r_1) = 1, \quad \epsilon(r_2) = 2, \quad \epsilon(r_5) = 1, \quad \epsilon(r_6) = 0, \\ \epsilon(r_3) = 0, \quad \epsilon(r_4) = 1, \quad \epsilon(r_7) = 2, \quad \epsilon(r_8) = 1, \end{aligned}$$

and if  $A$  and  $B$  are invertible, then they freely generate a free group.

**How to use Hilbert's Tenth Problem for Protocol 8, a private key cryptosystem with  $\text{Aut}(F)$ :**

In general, if Alice and Bob use the augmented ring

$$R := \mathbb{Z}[y_1, y_2, \dots, y_n],$$

with  $n \geq 2$ , consider 8 polynomials,

$$p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8 \in \mathbb{Z}[y_1, y_2, \dots, y_n]$$

and take privately an augmentation

$$\epsilon : \mathbb{Z}[y_1, y_2, \dots, y_n] \rightarrow \mathbb{Z},$$

such that

$$\begin{aligned} \epsilon(r_1) = 1, \quad \epsilon(r_2) = 2, \quad \epsilon(r_5) = 1, \quad \epsilon(r_6) = 0, \\ \epsilon(r_3) = 0, \quad \epsilon(r_4) = 1, \quad \epsilon(r_7) = 2, \quad \epsilon(r_8) = 1, \end{aligned}$$

then this augmentation induces a homomorphism

$$\epsilon^* : \text{GL}(2, R) \rightarrow \text{GL}(2, \mathbb{Z}),$$

with

$$\begin{aligned} A &:= \begin{pmatrix} p_1 & p_2 \\ p_3 & p_4 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = a \\ B &:= \begin{pmatrix} p_5 & p_6 \\ p_7 & p_8 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} = b. \end{aligned}$$

The matrices  $A$  and  $B$  generate a free subgroup of rank 2 because  $a$  and  $b$  generate a free subgroup of rank 2, see above.

To make use of Hilbert's Tenth Problem, the augmentation here is given by evaluating a point  $D = (d_1, d_2, \dots, d_n) \in \mathbb{Z}^n$ , such that

$$\epsilon^*(A) = \begin{pmatrix} p_1(D) & p_2(D) \\ p_3(D) & p_4(D) \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = a$$

and

$$\epsilon^*(B) = \begin{pmatrix} p_5(D) & p_6(D) \\ p_7(D) & p_8(D) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} = b.$$

To improve **Protocol 8**, a private key cryptosystem with  $Aut(F)$ , see the beginning of Chapter 7, Alice and Bob agree in addition on a subgroup  $J_W = \langle W \mid \ \rangle$ , with rank  $q$ , of  $J = \langle a, b \mid \ \rangle$ , with free generating set  $W = \{w_1, w_2, \dots, w_q\}$ . The set  $W$  is Nielsen reduced and the elements  $w_i$ ,  $1 \leq i \leq q$ , are words in  $\{a, b\}$ . Recall, that in the cryptosystem with  $Aut(F)$  the free group  $F$  is freely generated by the set  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ . Therefore, it is  $|W| = |X|$ .

The **public** knowledge for this modification extends to the augmented ring  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  with  $n \geq 2$ .

The **private** information for Alice and Bob extends to a point  $D \in \mathbb{Z}^n$ , the set  $\{a, b\}$  and the Nielsen reduced set  $W = \{w_1, w_2, \dots, w_q\}$  with  $w_i$  abstract words in  $\{a, b\}$ .

Note, that Alice and Bob do not fix 8 polynomials for the encryption and decryption procedure, respectively. For each encryption they can choose privately as an ephemeral key 8 polynomials,  $p_{1_j}, p_{2_j}, \dots, p_{8_j} \in R$ , to generate the matrices  $A_j$  and  $B_j$  in  $GL(2, R)$  with

$$A_j = \begin{pmatrix} p_{1_j} & p_{2_j} \\ p_{3_j} & p_{4_j} \end{pmatrix} \quad \text{and} \quad B_j = \begin{pmatrix} p_{5_j} & p_{6_j} \\ p_{7_j} & p_{8_j} \end{pmatrix}$$

and the property

$$p_{i_j}(D) = p_i(D) \quad \text{for all } i = 1, 2, \dots, 8 \quad (7.2)$$

and for  $D \in \mathbb{Z}^n$ , the common secret between Alice and Bob, and thus

$$\epsilon^*(A_j) = a \quad \text{and} \quad \epsilon^*(B_j) = b. \quad (7.3)$$

It is not necessary, that the decrypter knows which polynomials were needed if equality (7.2) and hence equality (7.3) holds.

The ciphertext  $C$  in the cryptosystem with  $Aut(F)$  is a sequence of ciphertext units  $c_i$ , written as words in  $X$ , it is,

$$C = c_1 \wr c_2 \wr \dots \wr c_z.$$

Alice and Bob identify  $x_i \in X$  with  $w_i \in W$ , for all  $i \in \{1, 2, \dots, q\}$ .

For **encryption** Alice writes the ciphertext  $C$ , which she generates as explained above, as a sequence of matrices in  $GL(2, R)$  with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ .

Firstly, she writes  $C$  as a sequence of words in  $W$ , that is,

$$C_W = \tilde{c}_1 \wr \tilde{c}_2 \wr \dots \wr \tilde{c}_z,$$

with  $\tilde{c}_i$ ,  $1 \leq i \leq z$  words in  $W$ . These words  $w_i$  are also abstract words in  $\{a, b\}$ , hence she gets the ciphertext

$$C' = c'_1 \wr c'_2 \wr \dots \wr c'_z,$$

with  $c'_i$ ,  $1 \leq i \leq z$ , abstract words in  $\{a, b\}$ . Secondly, Alice writes  $C'$  as a sequence of words in  $\{A_j, B_j\}$ , which is  $C'_{Hilbert}$  and means, instead of  $a$  she writes  $A_j$  and instead of  $b$  she writes  $B_j$ . It is

$$A_j = \begin{pmatrix} p_{1_j} & p_{2_j} \\ p_{3_j} & p_{4_j} \end{pmatrix} \quad \text{and} \quad B_j = \begin{pmatrix} p_{5_j} & p_{6_j} \\ p_{7_j} & p_{8_j} \end{pmatrix},$$

with  $p_{1_j}, p_{2_j}, \dots, p_{8_j} \in \mathbb{Z}[y_1, y_2, \dots, y_n]$  her ephemeral polynomials. Thus,

$$C'_{Hilbert} = \hat{c}_1 \wr \hat{c}_2 \wr \dots \wr \hat{c}_z$$

is the ciphertext  $C$  written as a sequence of matrices in  $GL(2, R)$  with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ . Hence, Alice sends each  $\hat{c}_i$ ,  $1 \leq i \leq z$ , as one matrix in  $GL(2, R)$ .

For **decryption** Bob uses the augmentation, which is given by evaluating  $p_{i_j}$ ,  $1 \leq i \leq 8$ , at the private point  $D$ , such that  $\epsilon^*(A_j) = a$  and  $\epsilon^*(B_j) = b$ . With this point  $D$  he is able to generate from  $C'_{Hilbert}$  the ciphertext version

$$\hat{C}' = \epsilon^*(\hat{c}_1) \wr \epsilon^*(\hat{c}_2) \wr \dots \wr \epsilon^*(\hat{c}_z) = \hat{c}'_1 \wr \hat{c}'_2 \wr \dots \wr \hat{c}'_z,$$

with  $\hat{c}'_i$ ,  $1 \leq i \leq z$ , matrices in  $SL(2, \mathbb{Z})$  which are words in  $\{a, b\}$ . With an algorithm (use for example the method described in Remark 4.3.12) to write the matrix  $\hat{c}'_i$  as an abstract word  $c'_i$  in  $\{a, b\}$  he gets

$$C' = c'_1 \wr c'_2 \wr \dots \wr c'_z.$$

Since Alice and Bob choose a Nielsen reduced set  $W = \{w_1, w_2, \dots, w_q\}$ ,  $w_j$  abstract words in  $\{a, b\}$ , Bob is now able to write each  $c'_i$ ,  $1 \leq i \leq z$ , as an abstract word in  $W$ , see Theorem 4.3.10 and Remark 4.3.11, and gets

$$C_W = \tilde{c}_1 \wr \tilde{c}_2 \wr \dots \wr \tilde{c}_z,$$

with  $\tilde{c}_i$ ,  $1 \leq i \leq z$  words in  $W$ .

Because of the identification of  $x_i \in X$  with  $w_i \in W$ , for all  $i \in \{1, 2, \dots, q\}$ , he is able to reconstruct the ciphertext

$$C = c_1 \wr c_2 \wr \dots \wr c_z,$$

with  $c_i$  words in  $X$ ,  $1 \leq i \leq z$ , as in the cryptosystem with  $Aut(F)$ . He finally decrypts the ciphertext as explained in the beginning of Chapter 7.

**Remark 7.3.8.** In an analogous way, also the modification in Section 7.1, with the ciphertext a reduced word  $C_{red}$  in  $X$ , can be improved with this approach. With the above procedure Alice generates of the reduced word  $C_{red}$  in  $X$ , one matrix  $M_{Hilbert}$  in  $GL(2, R)$ . Bob is able to reconstruct with the described procedure for decryption above from  $M_{Hilbert}$  the reduced word  $C_{red}$  and hence decrypts  $C_{red}$  as explained in Section 7.1. Remember that Alice has to sent additional information to Bob if the decryption is not only possible with  $C_{red}$ . Alice could also get the matrix  $M_{Hilbert}$  by multiplying the ciphertext matrices of the ciphertext  $C'_{Hilbert}$ , that is,

$$M_{Hilbert} = \prod_{i=1}^z \hat{c}_i,$$

with  $\hat{c}_i$  ciphertext units of the ciphertext  $C'_{Hilbert}$ .

**Security 7.3.9.** The security certification depends, in addition to Security 7.0.8, (for Remark 7.3.8 in addition to Security 7.1.6) on the unsolvability of Hilbert's Tenth Problem. Y. Matiyasevich proved in [Mat70] finally that there is no general algorithm which determines whether or not an integral polynomial in any number of variables has a zero. Therefore, for Eve, who sees just matrices in  $GL(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  and  $n \geq 2$ , it is hard to find the private key  $D$  of Alice and Bob. In addition the security is improved by the fact, that for each encryption Alice and Bob can take privately ephemeral matrices in  $GL(2, R)$  with the property that the common private point  $D \in \mathbb{Z}^n$  generates the correct matrices in  $PSL(2, \mathbb{Z})$ . This gives randomness to ciphertexts, which complicates attacks for Eve.

## 7.4. Chosen plaintext attacks on the cryptosystem with $Aut(F)$

In a chosen plaintext attack (see Section 1.1 or for instance also [BFKR15, Section 3.1]) Eve gives a blackbox, which does the encryption procedure, plaintexts of her choice and gets the corresponding ciphertexts. It is likely that Eve gives the blackbox a plaintext which obtains only one letter  $a_i$  of the alphabet  $A$ , to get information about the corresponding element  $u_i$ , it is,

$$\begin{aligned} S &= s_1 s_2 \cdots s_r \\ &= \underbrace{a_i a_i \cdots a_i}_{r \text{ times}} \end{aligned}$$

with  $a_i \in A$ . She gets the ciphertext

$$\begin{aligned} C &= f_\alpha(a_i) f_{h(\alpha)}(a_i) f_{h^2(\alpha)} \cdots (a_i) f_{h^{r-1}(\alpha)}(a_i) \\ &= c_1 c_2 \cdots c_r, \end{aligned}$$

with  $\alpha$  the starting seed to generate with the linear congruence generator  $h$  the used automorphisms  $f_j \in \mathcal{F}_{Aut}$  for encryption. If the ciphertext units have a similar structure, then an eavesdropper can maybe get information about the element  $u_i$  in the Nielsen reduced set  $U$  which corresponds to the plaintext letter  $a_i$ . This information could make the search in the Cayley graph more efficient.

**Example 7.4.1.** Eve gives the blackbox the plaintext

$$S = \text{YYYYYYYYY}.$$

We assume that the blackbox uses the same public and private parameters as in Example 7.0.7. The starting seed for the linear congruence generator  $h$  is  $x_1 = \bar{\alpha} = \overline{93}$  and the used automorphisms  $f_{x_1}, f_{x_2}, \dots, f_{x_8}$ , for encryption are computed as above in Example 7.0.7, hence the ciphertext is

$$\begin{aligned} C &= f_{x_1}(Y) f_{x_2}(Y) f_{x_3}(Y) \cdots f_{x_8}(Y) \\ &= c_1 c_2 c_3 \cdots c_8 \\ &= c(ad^2)^2 c^{-1} dc^{-1} ad^2 b^{-1} \wr (d^{-1} a^{-1} c^{-1})^2 dcadc^2 adb^{-1} d \wr (c^{-1} da)^2 ba^{-1} d^{-1} ab^3 da \wr \\ &\quad (ca^2)^2 db^{-1} a^3 dac a^2 d^{-1} a^{-3} b \wr (cb^{-1} a^{-1} b)^2 dc^{-2} b^{-1} a^{-1} bc^2 d^{-1} b \wr (ca^{-1})^2 dc^{-1} b^{-1} a^{-1} \wr \\ &\quad (c^{-1} a^{-1} ba^3)^2 dc^{-3} a^{-1} ba^3 c^3 d^{-1} ba^3 \wr (d^{-2} c)^2 d^{-1} b^{-1} d^{-1} a^{-1} b. \end{aligned}$$

The ciphertext units have a similar structure, which is

$$c_i = (w_i)^2 u_i,$$

for  $i = 2, 3, \dots, 8$ , with  $w_i, u_i$  words in  $X$ . The structure of  $c_1$  differs to the structures of the other ciphertext units.

Thus, an eavesdropper can assume that the element  $u_{10} \in U$  which corresponds to the letter  $a_{10} = Y$  is of the above form  $(w)^2 u$ , with  $w$  and  $u$  words in  $X$ .

In this example he is correct with this assumption because it is  $Y = a_{10} \hat{=} u_{10} = c^2 dab^{-1}$ , thus  $w = c$  and  $u = dab^{-1}$ .

To prevent this, Alice and Bob can agree privately on a way to change the used Nielsen reduced set  $U$ .

If we are in the situation (Section 7.1) in which the ciphertext  $C_{red}$  is a reduced word and the beginning and end of a ciphertext unit  $c_i$  is not marked it is more difficult to get information about a blackbox if words are used as inputs. If only one letter at the time is an input for the blackbox, then we get the same information as in the situation when the ciphertext units are identifiable in the unreduced ciphertext.

If we are in the situation (Section 7.2) in which the ciphertext  $C'$  is a matrix in  $\text{SL}(2, \mathbb{Q})$  this attack gives no information about the way the elements in the set  $U$  look like, because an eavesdropper, Eve, sees just matrices and she does not know which matrices are multiplied to get the ciphertext matrices  $c_i$ .

**Example 7.4.2.** If in Example 7.4.1 the used cryptosystem is chosen with the modification in which the ciphertext units are elements in  $\text{SL}(2, \mathbb{Q})$  and the blackbox used the parameters as in Example 7.2.4 (the same faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ ), then the ciphertext to the plaintext

$$S = \text{YYYYYYYY}$$

is

$$\begin{aligned} C' &= \varphi(f_{x_1}(\mathbf{Y}))\varphi(f_{x_2}(\mathbf{Y}))\varphi(f_{x_3}(\mathbf{Y})) \cdots \varphi(f_{x_8}(\mathbf{Y})) \\ &= \begin{pmatrix} 1362002520154399003411251 & 15571388221164541516505605 \\ -182887338329092260567748 & -2090899028244770708376289 \end{pmatrix} \\ &\quad \begin{pmatrix} -21110929144428898215300010362223029327 & 153141139922135745238345648793303290342 \\ -2918730152410756047224184025644787864 & 21172808624733035641035893652532742081 \end{pmatrix} \\ &\quad \begin{pmatrix} -264642814125471122620337910440849 & 1960560486141522671648480208507617 \\ -35535778402189873460069830975764 & 263260664038220168770908609864163 \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{-1854802475109324474047850088679642277698067443}{2} & \frac{12448535408701006001695125586831847496712873647}{4} \\ 124529831176821449990103535085350844783642259 & \frac{-835783881921361278554480278343662618989118139}{2} \end{pmatrix} \\ &\quad \begin{pmatrix} -32871793295402748701492250323594559338626411841 & 110309948059576753437092104294389254500749326264 \\ 4413968139903703503835378944511234293490078624 & -14812231017451052734153541914032788681367313857 \end{pmatrix} \\ &\quad \begin{pmatrix} 302888317565353 & 1032440955663986 \\ -40672482384904 & -138638350003831 \end{pmatrix} \\ &\quad \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \\ &\quad \begin{pmatrix} \frac{-2530644964961716069}{2} & \frac{16984462710235839663}{4} \\ -174957482575393773 & \frac{1174229842519006355}{2} \end{pmatrix}, \end{aligned}$$

with

$$\begin{aligned} m_1 &= -714125099204244682044465671027046210480583358681019787918242382304493 \\ m_2 &= 5290472191821927427878112760844761694412645622508690469944455927367053 \\ m_3 &= -95891478419140064071486038232134465359693838135391328268043601255684 \\ m_4 &= 710395420318448295524538695244804791684685771008576162902211439464607, \end{aligned}$$

see Appendix C.6. This gives no hint for the element  $u_{10} \hat{=} \mathbf{Y} = a_{10}$ .

If we are in the situation (Section 7.3) in which the ciphertext  $C'_{Hilbert}$  is a sequence of matrices in  $\text{GL}(2, R)$ , with  $R = [y_1, y_2, \dots, y_n]$  and  $n \in \mathbb{N}$ , this attack gives no hint for the elements in  $U$ . As in the situation in which the ciphertext is a sequence of matrices in  $\text{SL}(2, \mathbb{Q})$  an eavesdropper sees just matrices and she does not know which matrices are multiplied to get the ciphertext matrices  $\hat{c}_i$ , with  $i = 1, 2, \dots, r$ .

Conclusion concerning chosen plaintext attacks

If the ciphertext is given as a matrix, the system is secure against chosen plaintext attacks. If the ciphertext is a word in  $X$  it could be possible that an eavesdropper can get hints for the elements in  $U$  and hence the search for the primitive elements in the Cayley graph as well as the search for the automorphisms in  $\mathcal{F}_{Aut}$  could be performed in a more selective measure.

## 7.5. Chosen ciphertext attacks on the cryptosystem with $Aut(F)$

In a chosen ciphertext attack (see Section 1.1 or for instance also [BFKR15, Section 3.1]) Eve chooses ciphertexts and send these to a blackbox, she then gets the corresponding plaintexts back.

For example Eve gets to some parts of a given ciphertext the corresponding plaintext. These parts can be chosen by her.

Let

$$C = c_1 \wr c_2 \wr \cdots \wr c_z$$

be the ciphertext generated as explained in Chapter 7.

In this attack Eve gets for example the plaintext units  $s_j$  and  $s_{j+1}$  to the ciphertext units  $c_j$  and  $c_{j+1}$ , for a  $j$  with  $1 \leq j \leq z - 1$ .

Thus, she knows that

$$c_j \xrightarrow{\text{is decrypted to}} s_j = a_k, \tag{7.4}$$

$$c_{j+1} \xrightarrow{\text{is decrypted to}} s_{j+1} = a_\ell, \tag{7.5}$$

with  $k, \ell \in \{1, 2, \dots, N\}$ . In general, Eve gets no hint from (7.4) and (7.5) for the Nielsen reduced set  $U$ , which she needs for decryption (see Security 7.0.8). Maybe if she gets a lot of plaintext parts for a ciphertext and there are a few which decrypt the same alphabet letter, that means

$$c_{j_1} \xrightarrow{\text{is decrypted to}} s_{j_1} = a_k, \tag{7.6}$$

$$c_{j_2} \xrightarrow{\text{is decrypted to}} s_{j_2} = a_k, \tag{7.7}$$

$$\vdots \tag{7.8}$$

$$c_{j_s} \xrightarrow{\text{is decrypted to}} s_{j_s} = a_k \tag{7.9}$$

and the ciphertext units have a similar structure, for example

$$c_{j_i} = u_i(w_i)^2 v_i,$$

for  $i = 1, 2, \dots, s$  with  $u_i, v_i, w_i$  words in  $X$ , it is likely that she also assumes that  $u_k$  is of the structure

$$u_k = u'(w')^2 v',$$

with  $u', v', w'$  words in  $X$ . This is similar to the knowledge, which Eve could get by a chosen plaintext attack, see Section 7.4.

If Eve gets a candidate  $U'$  for  $U$  the information from (7.4) and (7.5) is useful for her. She

knows theoretically that

$$\begin{aligned} f_{x_j}(u_k) &= c_j \\ f_{x_{j+1}}(u_\ell) &= c_{j+1}. \end{aligned}$$

With this she can prove for all  $f_i \in \mathcal{F}_{\text{Aut}}$  if

$$\begin{aligned} f_i(u'_k) &= c_j \quad \text{or} \\ f_i(u'_\ell) &= c_{j+1}, \end{aligned}$$

with her  $u'_k$  and  $u'_\ell$  elements of her candidate set  $U'$ . Therefore, she can do the procedure to verify if  $U'$  is the used set  $U$  (both ordered by a lexicographical order) by Alice and Bob or to decrypt the ciphertext, respectively, in a more selective measure as in Security 7.0.8 described. Therefore, she proves if  $f_i(u'_s) = c_j$  only for two elements  $s = k$  and  $s = \ell$  (with  $k, \ell \in \{1, 2, \dots, N\}$ ) instead for all  $s \in \{1, 2, \dots, N\}$ . We assume that she knows the theoretical one-to-one correspondence  $a_j \mapsto u_j$ ,  $1 \leq j \leq N$ .

**Example 7.5.1.** Assume Eve is able to do a chosen ciphertext attack on the ciphertext  $C$  of Example 7.0.7, which is

$$\begin{aligned} C &= dc^{-1}d^{-1}a^{-1}d^{-2}a^{-1}c^{-1} \wr d^{-1}bcabd^{-1}a^{-1}cadb^{-1}d \wr \\ &\quad (ba^{-1}d^{-1})^2(a^{-1}d^{-1}c)^2 \wr (ca^2)^2b^{-1}a^3daca^2 \wr cb^{-1}a^{-1}bdc^{-2} \wr \\ &\quad bca(dc^{-1}b^{-1})^3ac^{-1} \wr a^{-1}(a^{-2}b^{-1})^2dc^{-3} \wr (ab^{-1})^3adc^{-1}d^2, \end{aligned}$$

and Eve knows

$$\begin{aligned} c_6 &= bca(dc^{-1}b^{-1})^3ac^{-1} \xrightarrow{\text{is decrypted to}} \mathbf{B} = a_{11} (\hat{=} u_{11}), \\ c_7 &= a^{-1}(a^{-2}b^{-1})^2dc^{-3} \xrightarrow{\text{is decrypted to}} \mathbf{O} = a_4 (\hat{=} u_4), \\ c_8 &= (ab^{-1})^3adc^{-1}d^2 \xrightarrow{\text{is decrypted to}} \mathbf{B} = a_{11} (\hat{=} u_{11}), \end{aligned}$$

remember, that she does not know how  $u_{11}$  or  $u_4$  are written as words in  $X = \{a, b, c, d\}$ . With this she could assume the following structures

$$\begin{aligned} u_4 &= u(w)^2v \\ u_{11} &= w'(v')^3u' \quad \text{or} \quad u_{11} = (\tilde{w})^3\tilde{u}, \end{aligned}$$

with  $u, w, v, w', v', u', \tilde{w}, \tilde{u}$  words in  $X$ . Note, that Alice and Bob used the following words for  $u_4$  and  $u_{11}$

$$u_4 = a^{-1}b \hat{=} \mathbf{B} = a_4 \quad \text{and} \quad u_{11} = a^{-1}d^3c^{-1} \hat{=} \mathbf{O} = a_{11}.$$

The assumed structure for  $u_4$  is wrong, only one of the assumed structures (which follows from  $c_6$ ) for  $u_{11}$  is correct, but Eve is not able to decide if the structure is correct or which structure is the correct one (for  $u_{11}$ ). Thus, the hint for elements of  $U$  is not as good as it could be in a chosen plaintext attack, see Section 7.5.

If she gets a candidate set  $U' = \{u'_1, u'_2, \dots, u'_{12}\}$  for the set  $U$ , used by Alice and Bob, it is likely that Eve calculates the element  $f_i(u'_4)$  for automorphisms  $f_i \in \mathcal{F}_{\text{Aut}}$ , this is a more selective measure for a search than without the knowledge of a chosen ciphertext attack (see Security 7.0.8). If  $f_i(u'_4) = c_6$  or  $f_i(u'_4) = c_8$  she gets a candidate for the automorphism  $f_{x_6}$  or  $f_{x_8}$ , respectively. If for all  $f_i \in \mathcal{F}_{\text{Aut}}$  it is  $c_6 \neq f_i(u'_4) \neq c_8$ , Eve knows that her set  $U'$  is not

correct.

Assume we are in the situation in which the ciphertext  $C_{red}$  is a reduced word in  $X$  and the beginning and end of each ciphertext unit  $c_i$  is not marked (see Section 7.1).

With a chosen ciphertext attack Eve gets information how the word  $C_{red}$  is assigned partly to the plaintext units. There could appear different cases:

1. If there are no cancellations for  $c_j$  and Alice knows which part of  $C_{red}$  corresponds to  $s_j$  then she knows  $c_j$  and knows where it ends and begins and hence where  $c_{j+1}$  begins and  $c_{j-1}$  ends. In general she does not know which number of  $\{1, 2, \dots, z\}$  is  $j$ . She now could act similar as in the unreduced ciphertext case above.
2. If there are cancellations for  $c_j$  it is not sure that Eve knows that. Maybe it is not necessary for Alice to give Bob such an additional information and hence Eve does not know if there are cancellations or not. Let

$$C_{red} \equiv w_1 \tilde{c}_j w_2$$

be the ciphertext with  $w_1, w_2$  words in  $X$  and  $c_j \equiv c_{j_1} \tilde{c}_j c_{j_2}$  with  $c_{j_1}, c_{j_2}$  words in  $X$  or the empty word. In a chosen ciphertext attack Eve gets the information

$$\tilde{c}_j \xrightarrow{\text{is decrypted to}} s_j = a_k (\hat{=} u_k)$$

for a  $k \in \{1, 2, \dots, N\}$ . It is  $c_j \equiv c_{j_1} \tilde{c}_j c_{j_2}$ , with  $c_{j_1}, c_{j_2}$  words in  $X$  or the empty word, but she does not know what  $c_{j_1}$  and  $c_{j_2}$  look like. Maybe she gets information about the structure of  $\tilde{u}_k$ , for  $u_k \equiv u_{k_1} \tilde{u}_k u_{k_2}$  with  $u_{k_1}, u_{k_2}$  words in  $X$ , because of the structure of  $\tilde{c}_j$ , with  $\tilde{c}_j$  is decrypted to  $\tilde{u}_k$ . In general she does not know which number of  $1, 2, \dots, z$  is  $j$ . It is known that  $c_j = f_{x_j}(u_k) = c_{j_1} \tilde{c}_j c_{j_2}$  for a  $f_{x_j} \in \mathcal{F}_{Aut}$ . Hence, if she gets a candidate set  $U' = \{u'_1, u'_2, \dots, u'_N\}$  for  $U$ , the Nielsen reduced set used by Alice and Bob, it is likely that she calculates the element  $f_i(u'_k)$  for automorphisms  $f_i \in \mathcal{F}_{Aut}$ . If  $f_i(u'_k) = u \tilde{c}_j v$  with  $u, v$  words in  $X$ , it is possible that  $f_i$  is the used automorphism  $f_{x_j}$  from Alice and Bob but this is not sure. It could also happen that  $f_i(u'_k) = u \tilde{c}_j v$  for more than one automorphism  $f_i \in \mathcal{F}_{Aut}$  or  $f_i(u'_k) = u \tilde{c}_j v$  but  $u'_k$  is not the element  $u_k$  from Alice and Bob. It is difficult for Eve to find the used set  $U$  and the automorphisms which were used by Alice and Bob, see also Security 7.1.6.

3. If the ciphertext is a sequence of different ciphertexts  $C_{red_i}$ ,  $1 \leq i \leq z'$ , given as words in  $X$ , and she gets a plaintext for example for one  $C_{red_j}$  she then knows of how many letters in  $U$ , and hence in  $A$  (the plaintext alphabet), the message  $C_{red_j} = c_{j_1} c_{j_2} \cdots c_{j_{z_j}}$  is written, that means, she knows  $|C_{red_j}|_U = z_j$ . In general she does not know where each ciphertext unit  $c_{j_k}$ ,  $1 \leq k \leq z_j$ , begins or ends, or if there are cancellations between the ciphertext units  $c_{j_k}$  and  $c_{j_{k-1}}$  or  $c_{j_k}$  and  $c_{j_{k+1}}$ . If the letter  $a_k$  is encrypted several times in  $C_{red_j}$ , then it could be that Eve gets a hint for the element  $u_k$ , see above (7.6) to (7.9), remember that there could occur cancellations, see 2.

Assume we are in the situation in which the ciphertext  $C'$  is a sequence of matrices in  $\text{SL}(2, \mathbb{Q})$ , see Section 7.2. Let

$$C' = W_1 W_2 \cdots W_z$$

be the ciphertext with  $W_i \in \text{SL}(2, \mathbb{Q})$ ,  $1 \leq i \leq z$ . With the chosen ciphertext attack, Eve gets



for example the information

$$\begin{aligned} W_j &\xrightarrow{\text{is decrypted to}} s_j = a_k(\hat{=}u_k), \\ W_{j+1} &\xrightarrow{\text{is decrypted to}} s_{j+1} = a_\ell(\hat{=}u_\ell), \end{aligned}$$

with  $1 \leq j \leq z-1$  and  $k, \ell \in \{1, 2, \dots, N\}$ . There is no hint for Eve for the used set  $M$ , to get the faithful representation  $\varphi$ , or the set  $U$ , used by Alice and Bob.

Only the brute force search described in Security 7.2.1 could be performed in a more selective measure, that means, Eve looks if

$$\varphi'(f_j(u'_k)) = W_j$$

instead if

$$W_j \in U'_{\varphi'(f_j)},$$

with  $f_j \in \mathcal{F}_{\text{Aut}}$ ,  $U' = \{u'_1, u'_2, \dots, u'_N\}$  her guessed set for  $U$  and  $\varphi'$  her guessed faithful representation, which she gets by her guessed set  $M'$  for  $M$ .

Even if she found a set  $U'$  and a set  $M'$ , such that

$$\varphi'(f_j(u'_k)) = W_j$$

she cannot be sure that  $U'$  and  $M'$  are the set used by Alice and Bob. There are a lot of sets, such that

$$\varphi'(f_j(u'_k)) = W_j$$

for some  $f_j \in \mathcal{F}_{\text{Aut}}$ .

Assume we are in the situation in which the ciphertext  $C_{\text{Hilbert}}$  is a sequence of matrices in  $\text{GL}(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ , see Section 7.3. This situation is analogous to the modification in which the ciphertext is a sequence of matrices in  $\text{SL}(2, \mathbb{Q})$ . Eve gets neither a hint for  $U$  nor  $A_j$  and  $B_j$  nor  $D$  nor  $W$ .

#### Conclusion concerning chosen ciphertext attacks

If the ciphertext is given as a matrix, the system is secure against chosen ciphertext attacks. If the ciphertext is a word in  $X$  it could be possible that an eavesdropper can get hints for the elements in  $U$  and hence the search for the primitive elements in the Cayley graph as well as the search for the automorphisms in  $\mathcal{F}_{\text{Aut}}$  could be performed in a more selective measure.



## Chapter 8

### Private key cryptosystem with $Aut(F_U)$ (Protocol 9)

In this chapter we introduce **Protocol 9**, which is a private key cryptosystem similar to **Protocol 8**. As **Protocol 8** it is based on combinatorial group theory. It uses a finitely generated free group  $F$ , a subgroup  $F_U$  of  $F$  with finite rank, a Nielsen reduced set and automorphisms of  $F_U$ . It differs to **Protocol 8** only in the way, that it uses automorphisms of the finitely generated subgroup  $F_U$  of  $F$  instead of automorphisms of the finitely generated free group  $F$ . The modifications of this cryptographic protocol use the ideas for the modifications of the **Protocol 8**. In the cryptographic protocol the ciphertext is a sequence of reduced words in  $X$  where the end of each ciphertext unit is marked and  $X$  is a free generating set for a free group  $F$  of finite rank. A modification is given where the ciphertext is only one reduced word in  $X$  instead of a sequence of words, in this case it is possible that additional information is needed for decryption, thus these is sent with the ciphertext if required. In the second modification a faithful representation from  $F$  into the special linear group  $SL(2, \mathbb{Q})$  is used, such that the ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$ . The third modification utilizes the negative solution of Hilbert's Tenth Problem. Instead of a presentation of the ciphertext as a sequence of matrices in  $SL(2, \mathbb{Q})$  the ciphertext is represented as a sequence of matrices in  $GL(2, R)$ , with  $R := \mathbb{Z}[y_1, y_2, \dots, y_n]$ , the integral polynomial ring in  $n \geq 2$  variables.

For the encryption of the plaintext different automorphisms are used for each plaintext unit, as in a One-Time-Pad (see for instance [MvOV97]). The automorphisms are out of a common set  $\mathcal{H}_{Aut} \subset Aut(H)$  (with  $H$  an abstract free group of finite rank). For decryption Bob needs to know which automorphisms of  $\mathcal{H}_{Aut}$  were used for the encryption procedure by Alice. For this choice of elements from  $\mathcal{H}_{Aut}$  regulations are needed. Therefore, Alice and Bob make use of a linear congruence generator with maximal periodic length as for **Protocol 8**. Hence, for linear congruence generators see Chapter 7.

Thus, we start this chapter with the description of **Protocol 9**. The modifications are explained next. We give for each cryptographic protocol in this chapter a security analysis and beside this we consider chosen plaintext attacks and chosen ciphertext attacks.

Now, we introduce **Protocol 9**. Before Alice and Bob are able to communicate with each other they have to make some arrangements.

#### Public Parameters

They first agree on the following public parameters.

1. A finitely generated free group  $F$  with free generating set  $X = \{x_1, x_2, \dots, x_q\}$  with  $q \geq 2$ .
2. A plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  with  $N \geq 2$ .

3. An abstract free group  $H = \langle U \mid \ \rangle$  with  $rank(H) = |A| = N$  and an abstract free generating set  $U = \{u_1, u_2, \dots, u_N\}$ , with  $u_i$ ,  $1 \leq i \leq N$ , abstract letters.
4. A subset  $\mathcal{H}_{Aut} := \{f_0, f_1, \dots, f_{2^{128}-1}\} \subset Aut(H)$  of automorphisms of  $H$ . It is  $f_i : H \rightarrow H$  and the  $f_i$ ,  $i = 0, 1, \dots, 2^{128} - 1$ , pairwise different, are generated with the help of the 0-1-sequence (of different length) and random numbers as described in Section 4.4. The set  $\mathcal{H}_{Aut}$  is part of the key space.
5. They agree on a linear congruence generator  $h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}}$  with a maximal period length.

### Private Parameters

Now, they agree on the private parameters.

1. Alice and Bob choose an explicit Nielsen reduced set  $U$  with  $N$  elements, which are words in  $X$ . Such systems  $U$  are easily to construct using Theorem 4.2.13 and Lemma 4.2.15 (see also [CgRR08] and [LS77]). Then  $F_U = \langle U \mid \ \rangle$  is a free subgroup of  $F$  with rank  $N$ , because of Theorem 4.2.13. It is  $\mathcal{U}_{Nred}$  the set of all minimal Nielsen reduced sets with  $N$  elements in  $F$ , which is part of the key space.
2. They use a one-to-one correspondence

$$A \rightarrow U$$

$$a_j \mapsto u_j \quad \text{for } j = 1, \dots, N.$$

3. Alice and Bob agree on an automorphism  $f_{\bar{\alpha}} \in \mathcal{H}_{Aut}$ , hence  $\alpha$  is the common secret starting point  $\alpha \in \{0, 1, \dots, 2^{128} - 1\}$ , with  $u_1 = \bar{\alpha} \in \mathbb{Z}_{2^{128}}$ , for the linear congruence generator. With this  $\alpha$  they are able to generate the sequence  $f_{u_1}, f_{u_2}, \dots, f_{u_z}$  (with  $z$  the number of the plaintext units, which are letters from  $A$ ) of automorphisms of the set  $\mathcal{H}_{Aut}$ , which they need for encryption and decryption, respectively.

**Remark 8.0.1.** If the explicit set  $U := \{u_1, u_2, \dots, u_N\}$ ,  $u_i$  words in  $X$ , is used, then  $F_U$  is a free subgroup of  $F$  and with the automorphism  $f_{u_j} \in \mathcal{H}_{Aut}$ , with  $f_{u_j} : F_U \rightarrow F_U$ , the set  $U_{f_{u_j}} = \{f_{u_j}(u_1), f_{u_j}(u_2), \dots, f_{u_j}(u_N)\}$  is generated, which is Nielsen equivalent to the set  $U$ .

**The key space:** The set  $\mathcal{U}_{Nred}$  of all minimal (with respect to a lexicographical order) Nielsen reduced sets of  $F$  with  $N$  elements. The set  $\mathcal{H}_{Aut}$  of  $2^{128}$  randomly chosen automorphisms of  $F_U$ .

### Private Key Cryptosystem

Now, we explain the private key cryptosystem and look carefully at the steps for Alice and Bob.

**Public knowledge:**  $F = \langle X \mid \ \rangle$ ,  $X = \{x_1, x_2, \dots, x_q\}$  with  $q \geq 2$ ; plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  with  $N \geq 2$ ; the set  $\mathcal{H}_{Aut}$ ; a linear congruence generator  $h$ .

#### **Encryption and Decryption Procedure:**

1. Alice and Bob agree privately on the private parameters: a set  $U \in \mathcal{U}_{Nred}$  and an automorphism  $f_{\bar{\alpha}} \in \mathcal{H}_{Aut}$ . They also know the one-to-one correspondence between  $U$  and  $A$ .

2. Alice wants to transmit the message

$$S = s_1 s_2 \cdots s_z, \quad z \geq 1,$$

with  $s_i \in A$  to Bob.

2.1. She generates with the linear congruence generator  $h$  and the knowledge of  $f_{\bar{\alpha}}$  the  $z$  automorphisms  $f_{u_1}, f_{u_2}, \dots, f_{u_z}$ , which she needs for encryption. It is  $u_1 = \bar{\alpha}$ ,  $u_2 = h(u_1)$ ,  $\dots$ ,  $u_z = h(u_{z-1})$ .

2.2. The encryption is as follows

$$\text{if } s_i = a_t \quad \text{then } s_i \mapsto c_i := f_{u_i}(a_t), \quad 1 \leq i \leq z, \quad 1 \leq t \leq N.$$

Recall that the one-to-one correspondence  $A \rightarrow U$  with  $a_j \mapsto u_j$ , for  $j = 1, 2, \dots, N$ , holds. The ciphertext

$$\begin{aligned} C &= f_{u_1}(s_1) f_{u_2}(s_2) \cdots f_{u_z}(s_z) \quad \text{with } s_i \hat{=} u_t \Leftrightarrow s_i = a_t \\ &= c_1 c_2 \cdots c_z \end{aligned}$$

is sent to Bob. As above  $c_j$  are called the ciphertext units and we do not perform cancellations between  $c_i$  and  $c_{i+1}$  and the end of each  $c_i$  is marked,  $1 \leq i \leq z-1$ , for example with the symbol “ $\gamma$ ”. On the one hand the ciphertext unit  $c_j$  can be seen as a word in  $U$ , because the set  $U_{f_{u_j}} = \{f_{u_j}(u_1), f_{u_j}(u_2), \dots, f_{u_j}(u_N)\}$  is Nielsen equivalent to  $U$  and  $f_{u_j}(s_j) \hat{=} f_{u_j}(u_k) =: c_j$ , for  $s_j = a_k$ , is an element in  $U_{f_{u_j}}$ . On the other hand it can be written as a word in  $X$ , because the explicit elements in  $U$  are words in  $X$  and so are the elements in the Nielsen equivalent set  $U_{f_{u_j}}$  to  $U$ .

3. Bob gets the ciphertext

$$C = c_1 c_2 \cdots c_z,$$

with  $c_j$ ,  $1 \leq j \leq z$ , words in  $X$ . He knows where each ciphertext unit  $c_j$  begins and ends. Hence, he gets the information that he has to use  $z$  automorphisms of  $F$  from the set  $\mathcal{H}_{Aut}$  for decryption. He has two possibilities for decryption.

3.1.a. With the knowledge of  $f_{\bar{\alpha}}$ , the set  $U = \{u_1, u_2, \dots, u_N\}$ , the linear congruence generator  $h$  and the number  $z$ , he computes for each automorphism  $f_{u_i}$ ,  $i = 1, 2, \dots, z$ , the set

$$U_{f_{u_i}} = \{f_{u_i}(u_1), f_{u_i}(u_2), \dots, f_{u_i}(u_N)\},$$

with  $f_{u_i}(u_j)$  written as a reduced word in  $X$ . Hence, with the one-to-one correspondence between  $U$  and  $A$ , he gets a one-to-one correspondence between the letters in the alphabet  $A$  and the words of the ciphertext depending on the automorphisms  $f_{u_i}$ . This is shown in Table 8.1 (page 190).

Table 8.1.: Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  corresponding to ciphertext alphabet  $U_{f_{u_i}}$  depending on the automorphisms  $f_{u_i}$

	$U_{f_{u_1}}$	$U_{f_{u_2}}$	$\dots$	$U_{f_{u_z}}$
$a_1$	$f_{u_1}(u_1)$	$f_{u_2}(u_1)$	$\dots$	$f_{u_z}(u_1)$
$a_2$	$f_{u_1}(u_2)$	$f_{u_2}(u_2)$	$\dots$	$f_{u_z}(u_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_N$	$f_{u_1}(u_N)$	$f_{u_2}(u_N)$	$\dots$	$f_{u_z}(u_N)$

With the knowledge of the Table 8.1 (page 190) the decryption is as follows

$$\text{if } c_i = f_{u_i}(u_t) \text{ then } c_i \mapsto s_i = a_t, \quad 1 \leq i \leq z, \quad 1 \leq t \leq N.$$

He generates the plaintext message

$$S = s_1 s_2 \dots s_z,$$

with  $s_i \in A$ , from Alice.

- 3.1.b. Bob knows the Nielsen reduced set  $U$ , hence with the algorithm in Theorem 4.3.10 he is able to write the elements  $c_i$  as words in  $U$ . With the knowledge of  $f_{\bar{\alpha}}$ , the set  $U = \{u_1, u_2, \dots, u_N\}$ , the linear congruence generator  $h$  and the number  $z$ , he gets the automorphism  $f_{u_i}$  which Alice used for encryption of  $c_i$ . Because of the fact that a one-to-one correspondence between  $A$  and  $U$  is used and the ciphertext unit  $c_i$  is an image of an element in  $U$  under the automorphism  $f_{u_i}$ , Bob knows with the automorphism  $f_{u_i}$  and the ciphertext unit  $c_i$  written as word in  $U$ , the plaintext letter  $a_j \in A$  which corresponds to the ciphertext unit  $c_i$ .

**Remark 8.0.2.** As soon as Alice and Bob agree on the starting seed automorphism and the Nielsen reduced set  $U$ , Bob is able to calculate the first columns of Table 8.1 (page 190) for decryption (he does not know how many columns he will need because he does not know yet how long the plaintext from Alice will be). If he gets the ciphertext  $C$  from Alice, he only has to do a search in the table to get the corresponding plaintext units to the ciphertext units. If columns are missing to decrypt the ciphertext, he calculates the missing columns. Thus, in Version 3.1.a. instead of Version 3.1.b. for decryption Bob is able to do calculations for decryption even before he knows the ciphertext.

**Remark 8.0.3.** The cryptosystem is a polyalphabetic system, that means, a word  $u_i \in U$ , and hence a letter  $a_i \in A$ , is encrypted differently at different positions in the plaintext, because different automorphisms are used during the encryption procedure for each ciphertext unit. Thus, for the ciphertext, a statistical frequency attack (see for instance [BFKR15]) over the frequency of words, which corresponds to letters in the plaintext alphabet, or groups of words, is useless.

We summarize **Protocol 9** in Table 8.2 (page 191).

Table 8.2.: Summary of **Protocol 9**: Private key cryptosystem with  $Aut(F_U)$

<b>Public Knowledge</b>	
$F = \langle X \mid \quad \rangle$ , $X = \{x_1, x_2, \dots, x_q\}$ , $q \geq 2$ ; plaintext alphabet $A = \{a_1, a_2, \dots, a_N\}$ , $N \geq 2$ ; abstract free group $H = \langle U \mid \quad \rangle$ , $U = \{u_1, u_2, \dots, u_N\}$ with $u_i$ abstract letters; set $\mathcal{H}_{Aut} \subset Aut(H)$ ; linear congruence generator $h$ of maximal periodic length.	
Alice	Bob
Private keys	
Explicit set $U = \{u_1, u_2, \dots, u_N\}$ with $u_i$ words in $X$ , $U \subset F$ Nielsen reduced set, $ U  = N$ ; seed $f_{\bar{\alpha}} \in \mathcal{F}_{Aut}$ , one-to-one correspondence $A \rightarrow U$ , $a_j \mapsto u_j$ .	
Encryption	
Choose message $S = s_1 s_2 \cdots s_z, \quad z \geq 1,$ with $s_i \in A$ . Calculate $u_1 = \bar{\alpha}, u_2 = h(u_1), \dots, u_z = h(u_{z-1})$ , obtain $f_{u_1}, f_{u_2}, \dots, f_{u_z}$ . Encryption procedure: if $s_i = a_t$ then $s_i \mapsto c_i := f_{u_i}(a_t)$ , $1 \leq i \leq z$ , $1 \leq t \leq N$ . Ciphertext: $C = f_{u_1}(s_1) f_{u_2}(s_2) \cdots f_{u_z}(s_z) = c_1 c_2 \cdots c_z$ , with $c_i$ written as words in $X$ .	$\xrightarrow{C=c_1 c_2 \cdots c_z}$
Decryption	
	Compute $z$ automorphisms: $u_1 = \bar{\alpha}, u_2 = h(u_1), \dots, u_z = h(u_{z-1})$ , obtain $f_{u_1}, f_{u_2}, \dots, f_{u_z}$ . <u>Two possibilities:</u> 1. For each $f_{u_i}$ , $i = 1, 2, \dots, z$ , compute $U_{f_{u_i}} = \{f_{u_i}(u_1), f_{u_i}(u_2), \dots, f_{u_i}(u_N)\}$ and get a table like Table 8.1 (page 190). (Decryption: Search in this table.) If $c_i = f_{u_i}(u_t)$ then $c_i \mapsto s_i = a_t$ , $1 \leq i \leq z$ , $1 \leq t \leq N$ . 2. Use Nielsen reduced set $U$ and an algorithm to write the ciphertext units $c_i$ (given as words in $X$ ) as words in $U$ . Together with the used automorphisms the ciphertext is decrypted correctly.  Reconstruct plaintext message $S = s_1 s_2 \cdots s_z$ , with $s_i \in A$ .

An example with decryption described as in 3.1.a. is given in Example 8.0.4. Another example with decryption as described in 3.1.b. is given in Appendix C.9.

**Example 8.0.4.** This example was executed in GAP. All details are given in Appendix C.8 Firstly, Alice and Bob agree on **public parameters**.

1. Let  $F$  be the free group on the free generating set  $X = \{x, y, z\}$ .
2. Let  $\tilde{A} = \{a_1, a_2, \dots, a_8\} = \{L, E, I, O, U, A, V, B\}$  be the plaintext alphabet.
3. Let  $H$  be the abstract free group of rank  $|\tilde{A}| = 8$  with free generating set  $U = \{u_1, u_2, \dots, u_8\}$ .
4. A set  $\mathcal{H}_{Aut} \subset Aut(H)$  is determined. In this example we give the automorphisms, which Alice and Bob use for encryption and decryption, respectively, just at the moment when they are needed.
5. The linear congruence generator with maximal periodic length is

$$h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}}$$

$$u \mapsto \overline{133}u + \overline{51}.$$

The **private parameters** for this example are the following:

1. Let  $F_U$  be the explicit finitely generated free group, which is generated with the free generating set  $U = \{u_1, u_2, \dots, u_8\}$  with words in  $X$ , for this example it is

$$u_1 := xyz, \quad u_2 := yzy^{-1}, \quad u_3 := x^{-1}zx^{-1}, \quad u_4 := y^{-1}x^2,$$

$$u_5 := z^{-1}xyx, \quad u_6 := z^{-1}yx^{-1}, \quad u_7 := x^3y, \quad u_8 := y^3z^{-2}.$$

The starting automorphism  $f_{u_1}$  is  $f_{\overline{23442}}$ , hence it is  $u_1 = \bar{\alpha} = \overline{23442}$ . It is known, that  $a_i \mapsto u_i, i = 1, 2, \dots, 12$ , for  $u_i \in U$  and  $a_i \in \tilde{A}$ , therefore

$$L \hat{=} u_1 = xyz, \quad E \hat{=} u_2 = yzy^{-1}, \quad I \hat{=} u_3 = x^{-1}zx^{-1}, \quad O \hat{=} u_4 = y^{-1}x^2,$$

$$U \hat{=} u_5 = z^{-1}xyx, \quad A \hat{=} u_6 = z^{-1}yx^{-1}, \quad V \hat{=} u_7 = x^3y, \quad B \hat{=} u_8 = y^3z^{-2}.$$

We now look at the encryption and decryption procedure for Alice and Bob.

2. With the above agreements **Alice** is able to encrypt her message

$$S = \text{LOVE}.$$

Her message is of length 4. She generates the ciphertext as follows:

- 2.1 First, she determines, with the help of the linear congruence generator  $h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}}$  with  $u \mapsto \overline{133}u + \overline{51}$  and the starting seed  $\bar{\alpha} = \overline{23442}$ , the four automorphisms  $f_{u_i} \in \mathcal{H}_{Aut}$ ,  $1 \leq i \leq 4$ , which she needs for encryption. It is

$$u_1 = \bar{\alpha} = \overline{23442}, \quad u_2 = h(u_1) = \overline{3117837},$$

$$u_3 = h(u_2) = \overline{414672372} \quad \text{and} \quad u_4 = h(u_3) = \overline{55151425527}.$$

The automorphisms are describable with the help of regular Nielsen transformations, it is  $f_{u_1} \hat{=} (N2)_{1.7} (N2)_{2.4} (N1)_5 (N2)_{7.8} [(N2)_{3.4}]^2 (N2)_{4.6} (N2)_{5.1} (N1)_7 (N2)_{6.3} (N2)_{8.1}$



$$(N2)_{7.4} (N1)_7 (N2)_{1.2} (N2)_{2.3} (N2)_{4.5},$$

$$f_{u_1} : H \rightarrow H$$

$$\begin{aligned} u_1 &\mapsto u_1 u_7 u_2 u_4, & u_5 &\mapsto u_5^{-1} u_1 u_7, \\ u_2 &\mapsto u_2 u_4 u_3 u_4^2, & u_6 &\mapsto u_6 u_3 u_4^2, \\ u_3 &\mapsto u_3 u_4^2, & u_7 &\mapsto u_6^{-1} u_4^{-1} u_7 u_8, \\ u_4 &\mapsto u_4 u_6 u_5^{-1} u_1 u_7, & u_8 &\mapsto u_8 u_1 u_7. \end{aligned}$$

$$f_{u_2} \hat{=} (N2)_{1.3} (N2)_{3.5} (N1)_2 (N1)_4 (N2)_{6.5} (N1)_1 [(N2)_{3.4}]^2 (N2)_{5.2} (N2)_{7.6} (N2)_{4.2} \\ (N2)_{2.8} (N2)_{8.4} (N1)_4 (N2)_{1.4} (N2)_{2.6} (N2)_{5.6} (N2)_{6.4} (N2)_{4.7},$$

$$f_{u_2} : H \rightarrow H$$

$$\begin{aligned} u_1 &\mapsto u_3^{-1} u_1^{-1} u_2 u_4, & u_5 &\mapsto u_5 u_2^{-1} u_6 u_5, \\ u_2 &\mapsto u_2^{-1} u_8 u_6 u_5, & u_6 &\mapsto u_6 u_5 u_2 u_4, \\ u_3 &\mapsto u_3 u_5 u_4^{-2}, & u_7 &\mapsto u_7 u_6 u_5, \\ u_4 &\mapsto u_2 u_4 u_7 u_6 u_5, & u_8 &\mapsto u_8 u_4^{-1} u_2^{-1}. \end{aligned}$$

$$f_{u_3} \hat{=} (N1)_2 (N1)_5 (N1)_8 (N2)_{6.3} (N2)_{3.7} (N2)_{1.2} [(N2)_{4.8}]^2 (N2)_{5.6} (N2)_{8.3} (N2)_{6.3} \\ (N1)_8 (N2)_{2.3} (N2)_{7.4} (N2)_{1.8} (N2)_{3.4},$$

$$f_{u_3} : H \rightarrow H$$

$$\begin{aligned} u_1 &\mapsto u_1 u_2^{-1} u_7^{-1} u_3^{-1} u_8, & u_5 &\mapsto u_5^{-1} u_6 u_3, \\ u_2 &\mapsto u_2^{-1} u_3 u_7, & u_6 &\mapsto u_6 u_3^2 u_7, \\ u_3 &\mapsto u_3 u_7 u_4 u_8^{-2}, & u_7 &\mapsto u_7 u_4 u_8^{-2}, \\ u_4 &\mapsto u_4 u_8^{-2}, & u_8 &\mapsto u_7^{-1} u_3^{-1} u_8. \end{aligned}$$

$$f_{u_4} \hat{=} (N1)_1 (N1)_3 (N1)_4 (N2)_{6.2} [(N2)_{8.2}]^3 (N2)_{2.3} (N2)_{3.4} (N2)_{5.2} (N2)_{7.4} (N2)_{1.3} \\ (N2)_{4.5} (N2)_{8.3} (N1)_1 (N1)_2 (N2)_{7.2} (N1)_3 (N2)_{2.3} (N2)_{3.5} (N2)_{6.1},$$

$$f_{u_4} : H \rightarrow H$$

$$\begin{aligned} u_1 &\mapsto u_4 u_3 u_1, & u_5 &\mapsto u_5 u_2 u_3^{-1}, \\ u_2 &\mapsto u_3 u_2^{-1} u_4 u_3, & u_6 &\mapsto u_6 u_2 u_4 u_3 u_1, \\ u_3 &\mapsto u_4 u_3 u_5 u_2 u_3^{-1}, & u_7 &\mapsto u_7 u_4^{-1} u_3 u_2^{-1}, \\ u_4 &\mapsto u_4^{-1} u_5 u_2 u_3^{-1}, & u_8 &\mapsto u_8 u_2^3 u_3^{-1} u_4^{-1}. \end{aligned}$$

2.2 Secondly, she encrypts her message. The ciphertext is

$$\begin{aligned} C &= f_{u_1}(\mathbf{L}) f_{u_2}(\mathbf{O}) f_{u_3}(\mathbf{V}) f_{u_4}(\mathbf{E}) \\ &= f_{u_1}(u_1) f_{u_2}(u_4) f_{u_3}(u_7) f_{u_4}(u_2) \\ &= u_1 u_7 u_2 u_4 \lambda u_2 u_4 u_7 u_6 u_5 \lambda u_7 u_4 u_8^{-2} \lambda u_3 u_2^{-1} u_4 u_3. \end{aligned}$$

The ciphertext  $C$  is a sequence of words in  $X$ , it is

$$\begin{aligned} C &= u_1 u_7 u_2 u_4 \lambda u_2 u_4 u_7 u_6 u_5 \lambda u_7 u_4 u_8^{-2} \lambda u_3 u_2^{-1} u_4 u_3 \\ &= x y z x^3 y^2 z y^{-2} x^2 \lambda y z y^{-2} x^5 y z^{-1} y x^{-1} z^{-1} x y x \lambda x^5 (z^2 y^{-3})^2 \lambda x^{-1} z x^{-1} y z^{-1} y^{-2} x z x^{-1}. \end{aligned}$$

3. **Bob** gets the ciphertext

$$C = xyzx^3y^2zy^{-2}x^2 \wr yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx \wr x^5(z^2y^{-3})^2 \wr x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1}$$

from Alice. Thus, he knows that he needs 4 automorphisms for decryption.

3.1. Bob knows the set  $U$ , the linear congruence generator  $h$  and the starting seed automorphism  $f_{\overline{23442}}$ . For decryption he uses tables like Table 8.1 (page 190).

Now, he is able to compute for each automorphism  $f_{u_i}$  the set  $U_{f_{u_i}}$ ,  $1 \leq i \leq 4$ , and to generate Table 8.3 (page 194) and Table 8.4 (page 194).

Table 8.3.: Correspondence: plaintext alphabet to ciphertext alphabet I

	$U_{f_{u_1}}$	$U_{f_{u_2}}$
L	$xyzx^3y^2zy^{-2}x^2$	$(xz^{-1})^2y^{-1}x^{-1}yzy^{-2}x^2$
E	$yzy^{-2}xzx^{-1}(y^{-1}x^2)^2$	$yz^{-1}y^2z^{-3}yx^{-1}z^{-1}xyx$
I	$x^{-1}zx^{-1}(y^{-1}x^2)^2$	$x^{-1}zx^{-1}z^{-1}x(yx^{-1})^2x^{-1}y$
O	$y^{-1}x^2z^{-1}yx^{-2}y^{-1}x^{-1}zxyzx^3y$	$yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx$
U	$x^{-1}y^{-1}x^{-1}zxyzx^3y$	$z^{-1}(xy)^2z^{-1}y^{-1}z^{-1}yx^{-1}z^{-1}xyx$
A	$z^{-1}yx^{-2}zx^{-1}(y^{-1}x^2)^2$	$z^{-1}yx^{-1}z^{-1}(xy)^2zy^{-2}x^2$
V	$xy^{-1}zx^{-2}yx^3y^4z^{-2}$	$x^3yz^{-1}yx^{-1}z^{-1}xyx$
B	$y^3z^{-2}xyzx^3y$	$y^3z^{-2}x^{-2}y^2z^{-1}y^{-1}$

Table 8.4.: Correspondence: plaintext alphabet to ciphertext alphabet II

	$U_{f_{u_3}}$	$U_{f_{u_4}}$
L	$xyzzyz^{-1}y^{-2}x^{-2}z^{-1}xy^3z^{-2}$	$y^{-1}xzyz$
E	$yz^{-1}y^{-1}x^{-1}zx^2y$	$x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1}$
I	$x^{-1}zx^4(z^2y^{-3})^2$	$y^{-1}xzx^{-1}z^{-1}(xy)^2zy^{-1}xz^{-1}x$
O	$y^{-1}x^2(z^2y^{-3})^2$	$x^{-2}yz^{-1}(xy)^2zy^{-1}xz^{-1}x$
U	$x^{-1}y^{-1}x^{-1}yx^{-2}zx^{-1}$	$z^{-1}(xy)^2zy^{-1}xz^{-1}x$
A	$z^{-1}y(x^{-2}z)^2x^2y$	$z^{-1}yx^{-1}yzy^{-2}xzyz$
V	$x^5(z^2y^{-3})^2$	$x^3yx^{-2}yx^{-1}zx^{-1}yz^{-1}y^{-1}$
B	$y^{-1}x^{-2}z^{-1}xy^3z^{-2}$	$y^3z^{-2}yz^3y^{-1}xz^{-1}x^{-1}y$

With these tables he is able to reconstruct the plaintext from Alice. He searches for the plaintext element  $s_i$  the ciphertext unit  $c_i$  in the column  $U_{f_{u_i}}$ ,  $1 \leq i \leq 4$ , and hence gets

the alphabet letter  $a_j = s_i$  for a  $j \in \{1, 2, \dots, 8\}$ . Therefore, he decrypts the ciphertext

$$C = xyzx^3y^2zy^{-2}x^2 \wr yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx \wr x^5(z^2y^{-3})^2 \wr x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1}$$

to the message

$$S = \text{LOVE.}$$

**Security 8.0.5.** An eavesdropper, Eve, intercepts the ciphertext

$$C = c_1c_2 \cdots c_z,$$

with  $c_i \in U_{f_{u_i}}$ ,  $1 \leq i \leq z$ . This is a **ciphertext only attack** also called known ciphertext attack (see Section 1.1 or for instance [BFKR15], [MvOV97] or [BNS10]). She knows where each ciphertext unit ends and begins, because there are no cancellations between  $c_i$  and  $c_{i+1}$ , and the end and the beginning of each  $c_i$  is marked. Alice and Bob generate the set  $U_{f_{u_i}}$  from the set  $U = \{u_1, u_2, \dots, u_N\}$  with the regular Nielsen transformation  $f_{u_i} \in \mathcal{H}_{Aut}$  on  $U$ , thus  $U_{f_{u_i}}$  is Nielsen equivalent to  $U$ , for all  $1 \leq i \leq z$ . This situation is visualized in Figure 8.1.

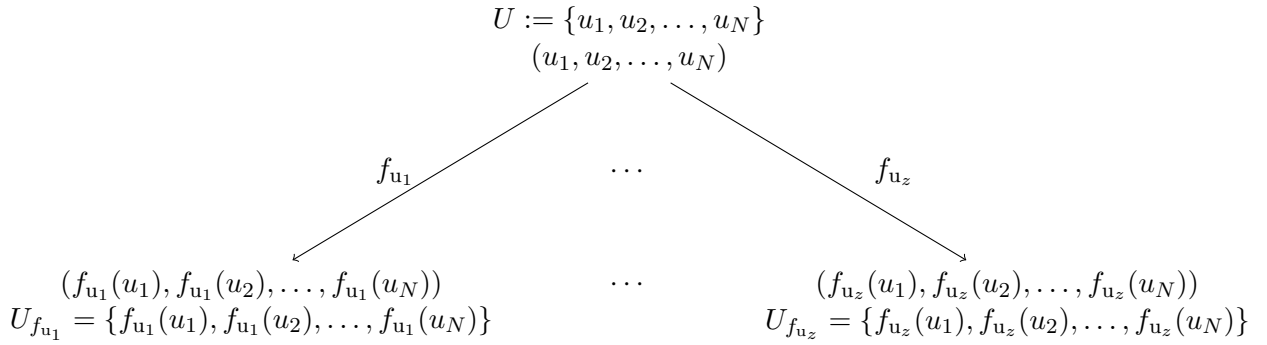


Figure 8.1.: Nielsen equivalent sets  $U_{f_{u_i}}$  to the set  $U$

The elements  $u_i$  are written as words in  $X$ . Therefore, the elements in  $U_{f_{u_i}}$  and hence the ciphertext units  $c_i$  are words in  $X$ . All sets  $U_{f_{u_i}}$  are Nielsen equivalent to  $U$  and therefore the ciphertext units are elements in the free subgroup  $F_U = \langle U \mid \rangle$  of  $F = \langle X \mid \rangle$ . Thus, the set of ciphertext units, that is,

$$\tilde{C} = \{c_1, c_2, \dots, c_z\}$$

generates a subgroup  $\langle \tilde{C} \rangle$  of  $F_U$ . Let  $\tilde{C}_{Nred}$  be a Nielsen reduced set of  $\tilde{C}$ , hence the group  $F_{\tilde{C}_{Nred}}$ , generated by  $\tilde{C}_{Nred}$ , is a free subgroup of  $F_U$ , that is,

$$F_{\tilde{C}_{Nred}} = \langle \tilde{C}_{Nred} \mid \rangle$$

is a subgroup of  $F_U = \langle U \mid \rangle$  and  $rank(F_{\tilde{C}_{Nred}}) \leq z$  because of Proposition 4.2.5. Let

$$L_1 := \max\{|c_i|_X \mid c_i \in \tilde{C}\}$$

be the maximum free length of the ciphertext units written as reduced words in  $X$ . After Remark 4.2.17 there exists an algorithm, such that Eve is able to generate  $\tilde{C}_{Nred}$  of  $\tilde{C}$  in a running time of  $\mathcal{O}(L_1^2 z^2)$ , with  $|\tilde{C}| = z$ .

To break the system, Eve needs to know the set  $U$ . If  $\langle \tilde{C} \rangle = F_U$  then  $\tilde{C}_{Nred} = U$  (and

$F_{\tilde{C}_{Nred}} = F_U$ ), because of Proposition 4.2.4 and the property, that Alice and Bob used a minimal Nielsen reduced set (with respect to a lexicographical order, see for instance Example 4.2.8), thus Eve is then able to identify the set  $U$  with the help of  $\tilde{C}_{Nred}$ .

In general it is  $\tilde{C}_{Nred} \neq U$  and hence  $F_{\tilde{C}_{Nred}}$  is not the free group  $F_U$ . Eve knows that  $F_{\tilde{C}_{Nred}}$  is a subgroup of  $F_U$ . Hence, she can use the set  $\tilde{C}_{Nred}$  to get candidates for  $U$ . For this she needs qualified elements  $\{w_1, w_2, \dots, w_\ell\}$ , with  $w_j$ ,  $1 \leq j \leq \ell$ , primitive elements in  $F$ . With these elements  $w_j$ ,  $1 \leq j \leq \ell$ , Eve can generate sets  $V_i := \tilde{C}_{Nred} \cup \{w_1, w_2, \dots, w_\ell\}$  with  $K := |V_i| \geq N$  and with Nielsen transformations she constructs Nielsen reduced sets  $V'_i$  to  $V_i$ . If  $|V'_i| = N$  then  $V'_i$  is a candidate for  $U$ . The running time is within  $\mathcal{O}(\lambda^2 K^2)$ , with  $\lambda := \max\left(\{|w_j|_X \mid j = 1, 2, \dots, \ell\} \cup \{|c'_i|_X \mid c'_i \in \tilde{C}_{Nred}\}\right) \leq L$ , to get the set  $V'_i$  from  $V_i$  with the algorithm [Ste89] (see Remark 4.2.17).

**Remark 8.0.6.** The ciphertext units are elements of  $F_U$ . Is it possible for Eve to get hints for the used elements in  $U$  analyzing the ciphertext units  $c_i$ ,  $1 \leq i \leq z$ ?

Eve intercepts ciphertext units  $c_i$ ,  $1 \leq i \leq z$ , written as words in  $X$ . Let  $c'_i$  be the ciphertext unit  $c_i$  written as a word in  $U$ , that is,

$$c'_i = u'_{i_1} u'_{i_2} \cdots u'_{i_{k_i}},$$

with  $1 \leq k_i \leq |c'_i|_X$ ,  $u'_{i_j} \in U^{\pm 1}$  and  $u'_{i_j} u'_{i_{j+1}} \neq 1$  for  $1 \leq j < k_i$ . It is  $k_i \leq |c'_i|_X \leq \sum_{j=1}^{k_i} |u'_{i_j}|_X$  because  $U$  is Nielsen reduced and hence each element in  $U$  has a stable letter (see Corollary 4.2.10 and Remark 4.2.11). It is likely that there are cancellations between the letters  $u'_{i_j}$  and  $u'_{i_{j+1}}$ , such that only a segment  $\tilde{u}'_{i_j}$  of  $u'_{i_j}$  written as a word in  $X$  (which contains its stable letter, see Remark 4.2.11) is visible in  $c_i$ . Eve is not able to identify which segment of  $c_i$  is a segment of  $u'_{i_j}$ , for all  $1 \leq j \leq k_i$ . She also does not know the free length of  $u'_{i_j}$  in  $X$  and which elements of  $U^{\pm 1}$  are used to write  $c'_i$ . Thus, it is difficult for her to get good hints for  $U$  by analyzing the ciphertext units  $c_i$ .

To make it more difficult for Eve Alice and Bob could change the set  $U$  frequently.

How can Eve find qualified elements  $w_j$  for the set  $V_i$ ?

Eve knows that Alice and Bob used a Nielsen reduced set  $U = \{u_1, u_2, \dots, u_N\}$  with words  $u_i$  in  $X$ ,  $1 \leq i \leq N$ . Each ciphertext unit  $c_i$ ,  $1 \leq i \leq z$ , can be written as a word in  $U$ , it is

$$c_i = u'_{i_1} u'_{i_2} \cdots u'_{i_{k_i}},$$

with  $1 \leq k_i \leq |c_i|_X$ ,  $u'_{i_j} \in U^{\pm 1}$  and  $u'_{i_j} u'_{i_{j+1}} \neq 1$  for  $1 \leq j < k_i$ . It is  $k_i \leq |c_i|_X$  because  $U$  is Nielsen reduced and hence each element in  $U$  has a stable letter (see Corollary 4.2.10 and Remark 4.2.11).

Let  $U'_{c_i} := \{u'_{i_1}, u'_{i_2}, \dots, u'_{i_{k_i}}\}$  be the set of all letters in  $U^{\pm 1}$  of the ciphertext unit  $c_i$ . Define  $U_{c_i} := U \cap U_{c_i}^{\pm 1}$  and

$$U_C := \bigcup_{i=1}^z U_{c_i},$$

it is  $U_C \subseteq U$  and the ciphertext units of  $C = c_1 c_2 \cdots c_z$  are words in  $U_C$ .

The set  $U$  is Nielsen reduced and hence, because of Corollary 4.2.12, it is  $|c_i|_X \geq |u'_{i_j}|_X$  for all  $1 \leq j \leq k_i$  and therefore

$$L_1 := \max\{|c_i|_X \mid c_i \in \tilde{C}\} \geq \max\{|u_j|_X \mid u_j \in U_C\}.$$

Thus, it is likely that each element of  $U$  can be found as a letter in at least one ciphertext unit  $c_i$

in  $\tilde{C}$ , hence it is likely that  $U_C = U$ . Therefore, Eve knows that the ball  $B(F, L_1)$  in the Cayley graph of  $F$  contains the set  $U_C$  and maybe although if  $U_C \neq U$  a whole basis for  $F_U$ . With this knowledge she searches for primitive elements in the ball  $B(F, L_1)$ . These primitive elements are her qualified elements  $w_j$ , which she needs to get the sets  $V_i$  and hence the candidates  $V'_i$  for  $U$ . Eve can be sure that she is able to write each ciphertext unit  $c_i$  with elements in her set  $V'_i$ , because each  $c_i$  can be written with  $\tilde{C}_{Nred}$  and all elements in  $\tilde{C}_{Nred}$  can be written with elements in  $V'_i$ , because  $\tilde{C}_{Nred} \subseteq V_i$ . Therefore, she can be sure, that there exists a set  $V'_i$ , such that  $U_C \subseteq V'_i$  with which she can decrypt the ciphertext  $C$  correctly if she gets the automorphism  $f_{u_i}$  for the ciphertext unit  $c_i$ .

To get all primitive elements in  $B(F, L_1)$  it took exponential time (see Proposition 4.2.21 and Theorem 4.2.22). Assume Eve gets a candidate  $V'$  for  $U$ . After construction, the words  $c_i$  can be written with letters of  $V'^{\pm 1}$ .

With the constructive membership problem, she writes the ciphertext units  $c_i$  with letters of  $V'^{\pm 1}$  (see Theorem 4.3.10 and Remark 4.3.11) and this gives her hints for the used automorphism  $f_{u_i}$ , which Alice used to get  $c_i$ . If she found an automorphism  $f_{u_i}$  corresponding to  $c_i$  she gets with the public linear congruence generator  $h$  the other automorphisms  $f_{u_j}$  for the ciphertext units  $c_j$ , and if she gets with these automorphisms and her set  $V'$  the correct elements  $c_j$ , Eve can be very sure that she found the correct set  $V'$  and hence she can also reconstruct the plaintext. For this she can use a statistical frequency attack (see for instance [BFKR15]) on her reconstructed plaintext given as words in  $V'$  over the frequency of words (or groups of words), which corresponds to letters in the plaintext alphabet.

**Remark 8.0.7.** Let

$$S = s_1 s_2 \cdots s_z,$$

$s_i \in A$  with  $A$  the set of plaintext letters, be the plaintext from Alice for Bob. Let

$$C = c_1 c_2 \cdots c_z$$

be the corresponding ciphertext to  $S$  and define  $\tilde{C} = \{c_1, c_2, \dots, c_z\}$ , the set of the ciphertext units.

Assume  $U_C \neq U$ , that means, there exists an element  $u_k \in U$ , such that  $u_k$  is not a letter in the ciphertext units  $c_i$ ,  $1 \leq i \leq z$ , of the ciphertext  $C$ .

If  $|u_k| \leq L_1$  then  $u_k$  is an element in the ball  $B(F, L_1)$  and Eve could find the correct set  $U$  which is used by Alice and Bob.

If  $|u_k| > L_1$  then  $u_k$  is not an element in the ball  $B(F, L_1)$  and Eve could not find the set  $U$ , but a set  $V'$ , with  $U_C \subset V'$ .

If we are in the situation in which  $U_C \neq U$  for the ciphertext  $C$ , Eve searches for sets  $V_i$  with  $V_i = \tilde{C}_{Nred} \cup W$ ,  $K = |V_i| \geq N$  and  $w_j \in W$  primitive elements in the ball  $B(F, L_1)$  of the Cayley graph from  $F$ ,  $L_1 = \max\{|c_i|_X \mid c_i \in \tilde{C}\}$ , and  $V'_i$  a Nielsen reduced set to  $V_i$  with  $|V'_i| = N$  as in Security 8.0.5. These sets  $V'_i$  are candidates for  $U$ .

With such a set  $V'_i$  she is able to generate the ciphertext units  $c_j$ , because of the way she generates these sets. Let  $V' = \{v_1, v_2, \dots, v_N\}$  be a Nielsen reduced set, which is one of Eve's candidates for  $U$ . Hence, with the algorithm, given in Theorem 4.3.10, which solves the constructive membership problem in free groups for subgroups with a Nielsen reduced set as generating set, Eve is able to write each element in  $\tilde{C}$  as a word of letters from the set  $V'^{\pm 1}$ .

Assume Eve writes the element  $c_i$  as  $c_i = v_{i_1} v_{i_2} \cdots v_{i_l}$ , with  $v_{i_j} \in V'^{\pm 1}$  for  $1 \leq j \leq l$ . This is a hint for the used automorphism by Alice. Therefore, Eve searches in the set  $\mathcal{H}_{Aut}$  of automorphism on  $U$  for such an automorphism, which applies an element  $u_k$ ,  $1 \leq k \leq N$ , to an element  $u_{i_1}^{\epsilon_1} u_{i_2}^{\epsilon_2} \cdots u_{i_l}^{\epsilon_l}$ , with  $i_a \in \{1, 2, \dots, N\}$  for  $1 \leq a \leq l$  and  $\epsilon_b \in \{1, -1\}$  for  $1 \leq b \leq l$ . If

$v_{i_p} = v_{i_q}$ ,  $p \neq q$ , then  $u_{i_p}^{\epsilon_p} = u_{i_q}^{\epsilon_q}$ , and  $v_{i_a}$  corresponds to  $u_{i_a}^{\epsilon_a}$ .

If Eve does not find such an automorphism in  $\mathcal{H}_{\text{Aut}}$ , she can be sure, that her set  $V'$  is false. If Eve finds such an automorphism  $f_\nu \in \mathcal{H}_{\text{Aut}}$ , with  $c_i \in V'_{f_\nu}$ , then it could be that Alice encrypted her plaintext letter  $s_i = a_k$ , for a  $k \in \{1, 2, \dots, N\}$ , of the plaintext  $S$  with this automorphism  $f_\nu$  to the ciphertext unit  $c_i$ . If this is true and if the one-to-one correspondence between  $A$  and  $U$  is public Eve knows that  $c_i$  corresponds to the element  $u_k$  and hence decrypts the plaintext letter  $a_k$ . If the one-to-one correspondence between  $A$  and  $U$  is not public, Eve can use a statistical frequency attack (see for instance [BFKR15]) on her reconstructed plaintext given as a sequence of words in  $U$  over the frequency of words (or groups of words), which corresponds to letters in the plaintext alphabet, to get the one-to-one correspondence.

Eve assumes that  $f_\nu$  is the automorphism  $f_{u_i}$ , which was used by Alice to encrypt  $s_i$  by  $c_i$ . She then proves if the other elements  $c_{i+\ell}$ ,  $1 \leq i + \ell \leq z$ , can be generated with her set  $V'$  and the corresponding automorphisms, which she is able to calculate, if she gets a correct automorphism  $f_\nu$ , which Alice used as  $f_{u_i}$ .

If  $1 \leq \ell \leq z - i$  she gets the automorphisms  $f_{h^\ell(\nu)}$  corresponding to  $c_{i+\ell}$  by calculating

$$h^\ell(\nu) = \underbrace{h(h(\dots h(h(\nu)) \dots))}_{\ell \text{ times } h},$$

remember that the linear congruence generator  $h$  is public.

If  $-i + 1 \leq \ell \leq -1$  she gets the automorphisms  $f_{h^\ell(\nu)}$  corresponding to  $c_{i+\ell}$  by calculating

$$h^\ell(\nu) = \underbrace{h^{-1}(h^{-1}(\dots h^{-1}(h^{-1}(\nu)) \dots))}_{|\ell| \text{ times } h^{-1}}.$$

The mapping  $h$  is a public linear congruence generator of maximal periodic length, hence bijective and therefore the inverse mapping of  $h$  exists.

There are two possibilities.

1.  $c_{i+\ell} \notin V'_{f_{h^\ell(\nu)}}$  :

It is possible that the automorphism  $f_\nu$  is not the used automorphism  $f_{u_i}$  by Alice to generate the ciphertext unit  $c_i$ . Then, Eve searches for another automorphism  $f_{\nu_1}$  in  $\mathcal{H}_{\text{Aut}}$ , which applies an element  $u_k$  to an element  $u_{i_1}^{\epsilon_1} u_{i_2}^{\epsilon_2} \dots u_{i_l}^{\epsilon_l}$ , with  $i_a \in \{1, 2, \dots, N\}$  for  $1 \leq a \leq l$  and  $\epsilon_b \in \{1, -1\}$  for  $1 \leq b \leq l$  (see above). If she finds such an automorphism she calculates the set  $V'_{f_{h^\ell(\nu_1)}}$ , which is Nielsen equivalent to the set  $V'$  under the automorphism  $f_{h^\ell(\nu_1)}$ , and searches for the element  $c_{i+\ell}$  in this set.

If  $c_{i+\ell} \in V'_{f_{h^\ell(\nu_1)}}$ , Eve is then in case 2. below. If  $c_{i+\ell} \notin V'_{f_{h^\ell(\nu_1)}}$  she is again in this case 1. and tries to find another automorphism  $f_{\nu_2}$  in  $\mathcal{H}_{\text{Aut}}$ , which applies an element  $u_k$  to an element  $u_{i_1}^{\epsilon_1} u_{i_2}^{\epsilon_2} \dots u_{i_l}^{\epsilon_l}$ , with  $i_a \in \{1, 2, \dots, N\}$  for  $1 \leq a \leq l$  and  $\epsilon_b \in \{1, -1\}$  for  $1 \leq b \leq l$  (see above).

If she tried all possible automorphisms and got every time the case 1. then she has definitely a wrong set  $V'$ , that means it is  $U_C \not\subset V'$ . She then has to change her candidate  $V'$  for  $U$ .

2.  $c_{i+1} \in V'_{f_{h(\nu)}}$  :

If she is able to generate each element  $c_j \in \tilde{C}$  with this set  $V'$  and the automorphisms which she gets with her seed automorphism  $f_\nu \in \mathcal{H}_{\text{Aut}}$  and the linear congruence generator  $h$  then it is very likely that she found the correct automorphism  $f_\nu$  which is  $f_{u_i}$  and a set  $V'$  with  $U_C \subset V'$ . With these automorphisms she is able to identify which elements of  $V'$  correspond to the set  $U_C$  and which to the set  $W$ , it is  $V' = U_C \cup W$ . It is possible,

that the elements in  $W$  are not elements in  $U$ . If she is not able to generate each element  $c_j \in \tilde{C}$  with the automorphisms which follows from her automorphism  $f_{h(u)}$  she then is in case 1. above.

Assume Eve knows a set  $V'$  with  $U \neq V' = U_C \cup W$ , thus she knows the sets  $U_C \subset U$  and  $W$ , hence she knows which elements of  $U$  are for sure in her set  $V'$ . She also knows the automorphisms  $f_{u_t}$ , which Alice used to get the ciphertext units  $c_t$  for the plaintext units  $s_t$ , for all  $1 \leq t \leq z$ .

If Eve intercepts now the next ciphertext  $C_1 = c_{z+1}c_{z+2} \cdots c_{z_1}$ , she is able to calculate the automorphism  $f_{u_{z+1}}$ , because she knows  $f_{u_z}$  and it is  $h(u_z) = u_{z+1}$ , with  $h$  the used linear congruence generator. Eve calculates the set  $V'_{f_{u_{z+1}}}$  and searches for the element  $c_{z+1}$  in this set. There occur two cases.

1. If  $c_{z+1} \in V'_{f_{u_{z+1}}}$  then  $c_{z+1}$  is a word in  $U_C$  or it is a word in  $V'$  which has at least one letter in the set  $W^{\pm 1}$ . Eve is able to decide if a letter of  $W^{\pm 1}$  was used to write  $c_{z+1}$ . If the second case arises, Eve knows that these letters of  $W^{\pm 1}$  are candidates for the set  $U$ . Hence, she gets closer to the set  $U$ . The actual set  $V'$  could be  $U$ . It is likely that she decrypts the ciphertext unit  $c_{z+1}$  correctly.
2. If  $c_{z+1} \notin V'_{f_{u_{z+1}}}$  then  $c_{z+1}$  has at least one letter used which is not in  $U_C$  and also not in  $W$ , hence there are elements in  $W$  which do not belong to the set  $U$  and the actual set  $V'$  is not  $U$ . Eve gets one of the following situations:
  - a) The ciphertext  $c_{z+1}$  is written with elements of  $U \setminus V'$ . She then knows, that no element of  $W$  is a letter for  $c_{z+1}$ . She knows the automorphisms and hence only the elements  $u_k$  who map to words which are written with elements not in  $U_C$  are possible for the ciphertext  $c_{z+1}$ . She knows that  $c_{z+1}$  is written with up to  $\min\{|W|, |c_{z+1}|\}$  elements of  $U \setminus U_C$ . Maybe this information is enough to identify the corresponded letter to  $c_{z+1}$ . Eve could also try to get correct letters for  $c_{z+1}$  which are in  $U \setminus U_C$ , therefore she uses other elements for  $W$  as before. These elements are primitive elements in a ball  $B(F, |c_{z+1}|)$  of the Cayley graph from the free group  $F$ . She could use the information of the words which are written with elements not in  $U_C$  and the ciphertext unit  $c_{z+1}$  to get new candidates for  $W$  and hence Eve comes closer to the correct set  $U$ . It is not necessary to know the whole set  $U$  to decrypt the ciphertext unit, if the set  $U_C$  gives enough information to identify  $u_k$  for the ciphertext unit  $u_{z+1}$  if the automorphism  $f_{u_{z+1}}$  is known.
  - b) The ciphertext unit  $c_{z+1}$  is written with letters in  $V'^{\pm 1}$  and letters not in  $V'^{\pm 1}$ . Eve searches for the element  $c_{z+1}$  in the set  $V'_{f_{u_{z+1}}}$ , because  $c_{z+1}$  has letters which are elements of  $V'^{\pm 1}$  there are elements in  $V'_{f_{u_{z+1}}}$  which match partly with sequences in  $c_{z+1}$ . Sequences in  $c_{z+1}$  which are different must occur to elements  $u_\ell \in U$  which are not in Eve's set  $V'$ . Eve is able to identify at least sequences with stable letters for these elements  $u_\ell$ . Maybe it is enough to search for an element in  $V'_{f_{u_{z+1}}}$  which matches partly to  $c_{z+1}$  to get the correct plaintext letter for  $c_{z+1}$ , this element  $c_{z+1} = f_{u_{z+1}}(u_t)$  (if  $s_{z+1} = a_t$ ) must be written in  $U$  with some letters which are not yet in  $V'$ . For these letters she can identify sequences in  $c_{z+1}$  and hence gets hints for the elements in  $U$ , which are not yet in  $V'$ . It is not necessary to know the whole set  $U$ , because maybe the set  $U_C$  and the automorphism  $f_{u_{z+1}}$  give enough information to identify the correct plaintext letter to the element  $c_{z+1}$ . If this is possible she also gets hints for elements in the set  $U \subset U_C$ .

This is analogous for the other ciphertext elements in the ciphertext  $C_1 = c_{z+1}c_{z+2} \cdots c_{z_1}$ .

The main security certification depends on the fact, that for a single subset  $V$  of  $F_U$  with  $K$  elements Eve finds a Nielsen reduced set in the running time  $\mathcal{O}(\lambda^2 K^2)$ , with  $\lambda$  the maximum over the free length of the elements in the subset  $V$  with  $K$  primitive elements, but she has to test all possible subsets of  $K$  elements for which she needs exponential running time, because the number of primitive elements grows exponentially with the free length, see Proposition 4.2.21 and Theorem 4.2.22. She searches in a ball  $B(F, L_1)$ , with  $L_1 = \max\{|c_i| \mid c_i \in \tilde{C}\}$  for these primitive elements.

A subset of  $V$  is also known, it is  $\tilde{C}_{Nred} \subset V$  but she has to put all other primitive elements to this set and proves if  $V'$ , which is Nielsen reduced to  $V$ , is of order  $N$  and hence a candidate for  $U$ .

To verify the set  $V'$  as  $U$  or to find the automorphisms for encryption and decryption, respectively, and hence decrypt the message, Eve gets hints for the used automorphisms in  $\mathcal{H}_{Aut}$  by solving the constructive membership problem of the elements  $c_i$  of the ciphertext with her candidate  $V'$  for  $U$ , which is Nielsen reduced. Thus, this is not only a brute force search.

**Remark 8.0.8.** If the set  $\mathcal{H}_{Aut}$  is private Eve could write with the correct set  $V' = U$  and the constructive membership problem each element  $c_i$  as a word in  $U$ , but she does not know to which element  $u_k \in U$  the ciphertext unit  $c_i$  corresponds. If the set of automorphisms  $Aut(H)$  is not restricted by the set  $\mathcal{H}_{Aut}$ , then every correspondence between  $c_i$  and  $u_k$  (and hence plaintext unit  $a_k$ ) is equivalently likely. A statistical frequency attack (see for instance [BFKR15]) is useless, because even if  $s_i = s_j$ ,  $i \neq j$ , it is  $c_i \neq c_j$  as words both in  $X$  and  $U$ . Alice and Bob made the set  $\mathcal{H}_{Aut}$  public because they are then able to change the automorphisms without a private meeting, see Remark 7.0.3. If they use Variation 7.0.4 they have a public part of the set  $\mathcal{H}_{Aut}$  and a private part, hence they make an attack for Eve more difficult than in the situation when the set  $\mathcal{F}_{Aut}$  is completely public.

**Remark 8.0.9.** If Alice and Bob used an arbitrary Nielsen reduced set  $U$  and not a minimal Nielsen reduced set concerning to a lexicographical order, than Eve gets for  $V'$  much more sets, see Example 4.2.16. She gets also sets  $V'$  which are the set  $U$  but with permuted order. In this case she has to test all permuted sets and not only the minimal set (concerning to a lexicographical order), thus she gets  $N! - 1$  more sets for each  $V'$  to test with the automorphisms. There are also Nielsen reduced sets, which generate the same group but differ not only in the permutation order but also in some elements, for example  $\{y^2, y^{-1}xy\}$  and  $\{y^2, yxy^{-1}\}$  generate the same free group (see Example 4.2.16).

The security certification can be improved by the next three modifications, which are explained in Section 8.1, Section 8.2 and Section 8.3.

## 8.1. Modification with the ciphertext a reduced word for the cryptosystem with $Aut(F_U)$

As in Section 7.1 the reduced word  $C_{red}$  of the ciphertext  $C$  is sent instead of  $C$ , hence there are no parts with  $x_k x_k^{-1}$ , for a  $1 \leq k \leq q$ , which only occur in  $C$  if  $x_k$  is the last letter of  $c_j$  and  $x_k^{-1}$  is the first letter of  $c_{j+1}$ . The beginning and end of each ciphertext unit  $c_i$  is not marked. Let

$$C_{red} = x'_1 x'_2 \cdots x'_{L'},$$

with  $x'_i \in X^{\pm 1}$ ,  $1 \leq i \leq L'$ , be the reduced ciphertext from Alice for Bob. This word can also



be seen as a reduced word in  $U$ , that means

$$\hat{C}_{red} = \hat{u}_1 \hat{u}_2 \cdots \hat{u}_\ell,$$

with  $\hat{u}_i \in U^{\pm 1}$ ,  $1 \leq i \leq \ell$ . It is  $\ell \leq L'$  because of Corollary 4.2.12.

Alice sends the word  $C_{red}$  with letters in  $X^{\pm 1}$  as ciphertext to Bob.

The method how Bob decrypts the ciphertext  $C_{red}$  with the help of tables like Table 8.1 (page 190) and the kind of additional information which Alice has to give to Bob, if the decryption is not unique, is analogous to the description in Section 7.1, in which the cryptosystem uses  $Aut(F)$ .

**Example 8.1.1.** In Example 8.0.4 the ciphertext is

$$C = xyzx^3y^2zy^{-2}x^2 \wr yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx \wr x^5(z^2y^{-3})^2 \wr x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1}.$$

The reduced ciphertext is

$$C_{red} = xyzx^3y(yzy^{-2}x^2)^2x^3yz^{-1}yx^{-1}z^{-1}xyx^6(z^2y^{-3})^2x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1}.$$

In this example are no reductions between the ciphertext units and all ciphertext units are uniquely identifiable in the corresponding set  $U_{f_{u_i}}$ . Thus, no additional information is needed from Alice for decryption. In the first moment Bob does not know how many sets  $U_{f_{u_i}}$  and hence how big the table (a table like Table 8.1 (page 190)) will be, which he needs for decryption, but he knows the set  $U$ , the starting automorphism  $f_{\bar{\alpha}}$ , the used set  $\mathcal{H}_{Aut}$  and the linear congruence generator  $h$ , hence he is able to calculate the required sets  $U_{f_{u_i}}$  for the tables. The tables for this example are given in Example 8.0.4, see Table 8.3 (page 194) and Table 8.4 (page 194).

Bob could also use instead of tables (like Table 8.1 (page 190)), which store words in  $X$ , the algorithm which solves the constructive membership problem (see Theorem 4.3.10) to reconstruct the message from Alice. He knows the Nielsen reduced set  $U$  and hence he is able to write the ciphertext  $C_{red}$  as a word in  $U$ . Thus, he gets from  $C_{red}$  the word  $\hat{C}_{red}$  written in  $U$ . With the starting automorphism  $f_{\bar{\alpha}}$ , the used set  $\mathcal{H}_{Aut}$  and the linear congruence generator  $h$ , Bob is able to calculate the used automorphisms  $f_{u_i}$ . He then gets tables, like Table 8.5 (page 201), which store words in  $U$  and give a correspondence between the alphabet letters in the set  $A$  and the images of the automorphisms  $f_{u_i} \in \mathcal{H}_{Aut}$ , because of the one-to-one correspondence between  $A$  and  $U$ .

Table 8.5.: Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  corresponding to the images of the automorphisms  $f(u_i)$

	$f_{u_1}$	$f_{u_2}$	$\cdots$	$f_{u_z}$
$a_1 \hat{=} u_1$	$f_{u_1}(u_1)$	$f_{u_2}(u_1)$	$\cdots$	$f_{u_z}(u_1)$
$a_2 \hat{=} u_2$	$f_{u_1}(u_2)$	$f_{u_2}(u_2)$	$\cdots$	$f_{u_z}(u_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_N \hat{=} u_N$	$f_{u_1}(u_N)$	$f_{u_2}(u_N)$	$\cdots$	$f_{u_z}(u_N)$

With these tables and additional information from Alice, if required, he is able to decrypt the ciphertext correctly. The additional information and the method for decryption is analogous to the description in Section 7.1, in which the cryptosystem uses  $Aut(F)$ .

Alice and Bob agree if the additional information is for  $C_{red}$  or  $\hat{C}_{red}$ . Or Alice gives this information also in her additional information. She can use instead of  $(i, d, \tilde{w})$ , see above in Section 7.1, the additional information  $(i, d, \tilde{w}, X)$  if the information is about  $C_{red}$  and the others entries have the same meaning as before. Or  $(i, d, \tilde{w}, U)$  if the information is about  $\hat{C}_{red}$  and the other entries are analogous for words in  $U$ .

**Example 8.1.2.** If Bob does the decryption with using  $\hat{C}_{red}$  and tables like Table 8.5 (page 201), he gets for

$$C_{red} = xyzx^3y(yzy^{-2}x^2)^2x^3yz^{-1}yx^{-1}z^{-1}xyx^6(z^2y^{-3})^2x^{-1}zx^{-1}yz^{-1}y^{-2}zx^{-1},$$

from Example 8.1.1, the ciphertext

$$\hat{C}_{red} = u_1u_7(u_2u_4)^2u_7u_6u_5u_7u_4u_8^{-2}u_3u_2^{-1}u_4u_3.$$

The required steps can be taken from the tables in Appendix C.9.

With the starting automorphism  $f_{u_1}$ , which is  $f_{\overline{23442}}$  for this example (Example 8.0.4), and the other common information between Alice and Bob, he is able to get step by step the Table 8.6 (page 202) and to decrypt also step by step the ciphertext  $\hat{C}_{red}$ , with the help of the columns of the Table 8.6 (page 202), correctly. He needs no additional information from Alice, because no analogous situation as described in Remark 7.1.3 appears and hence the decryption is uniquely possible for Bob without additional information.

Table 8.6.: Plaintext alphabet  $A = \{L, E, I, O, U, A, V, B\}$  corresponding to the images of the automorphisms  $f_{u_i}$ ,  $i = 1, 2, 3, 4$

	$f_{u_1}$	$f_{u_2}$	$f_{u_3}$	$f_{u_4}$
$L \hat{=} u_1$	$u_1u_7u_2u_4$	$u_3^{-1}u_1^{-1}u_2u_4$	$u_1u_2^{-1}u_7^{-1}u_3^{-1}u_8$	$u_4u_3u_1$
$E \hat{=} u_2$	$u_2u_4u_3u_4^2$	$u_2^{-1}u_8u_6u_5$	$u_2^{-1}u_3u_7$	$u_3u_2^{-1}u_4u_3$
$I \hat{=} u_3$	$u_3u_4^2$	$u_3u_5u_4^{-2}$	$u_3u_7u_4u_8^{-2}$	$u_4u_3u_5u_2u_3^{-1}$
$O \hat{=} u_4$	$u_4u_6u_5^{-1}u_1u_7$	$u_2u_4u_7u_6u_5$	$u_4u_8^{-2}$	$u_4^{-1}u_5u_2u_3^{-1}$
$U \hat{=} u_5$	$u_5^{-1}u_1u_7$	$u_5u_2^{-1}u_6u_5$	$u_5^{-1}u_6u_3$	$u_5u_2u_3^{-1}$
$A \hat{=} u_6$	$u_6u_3u_4^2$	$u_6u_5u_2u_4$	$u_6u_3^2u_7$	$u_6u_2u_4u_3u_1$
$V \hat{=} u_7$	$u_6^{-1}u_4^{-1}u_7u_8$	$u_7u_6u_5$	$u_7u_4u_8^{-2}$	$u_7u_4^{-1}u_3u_2^{-1}$
$B \hat{=} u_8$	$u_8u_1u_7$	$u_8u_4^{-1}u_2^{-1}$	$u_7^{-1}u_3^{-1}u_8$	$u_8u_2^3u_3^{-1}u_4^{-1}$

**Remark 8.1.3.** Which is Eve's information of the additional information from Alice for Bob? These are similar to the information in the cryptosystem which uses  $Aut(F)$ , see Remark 7.1.5. Eve knows from

1.  $(i, |c_i|, 1, X)$ , that there is at least one element in  $U_{f_{u_i}}$  of free length  $|c_i|$ . After Corollary 4.2.12 Eve knows that  $|c_i|_X \geq |f_{u_i}(u_t)|_U$  for all  $1 \leq t \leq N$ , hence all automorphisms of  $\mathcal{H}_{\text{Aut}}$  with  $|f_{u_i}(u_t)|_U > |c_i|_X$  for all  $1 \leq t \leq N$  are not used by Alice to encrypt  $s_i$  to  $c_i$ , but she does not know if this element is completely visibly in  $C_{\text{red}}$  or  $C_{\text{red}}^{(i)}$ ;
2.  $(i, |c'_i|_X, w, X)$ , with  $c'_i$  an initial segment of  $c_i$  and  $w$  an segment of  $c_i$  ( $c_i \equiv c'_i w \tilde{w}$  with  $\tilde{w}$  word in  $X$  or the empty word), that  $|c_i| \geq |c'_i| + |w|$  and the first  $|c'_i|$  elements of  $C_{\text{red}}^{(i)}$  are the first  $|c'_i|$  elements of  $c_i$  and after these elements comes the word  $w$  in  $c_i$ . If  $(i, |c'_i|_X, w1, X)$ , then  $|c_i| = |c'_i| + |w|$  and Eve gets the analogous information for the automorphism  $f_{u_i}$  as above in 1., but she does not know, if this word  $w$  is visible in  $C_{\text{red}}$  or  $C_{\text{red}}^{(i)}$ ;
3.  $(i, 0, c'_i, X)$ , with  $c'_i$  the initial segment of  $c_i$ , that the word  $c_i$  is completely canceled in  $C_{\text{red}}$ . She also knows that  $c'_i$  is an initial segment of  $c_i$ , thus  $|c_i| \geq |c'_i|$ , it is not necessary that  $c'_i = c_i$ ;
4.  $(i, |c_i|_U, 1, U)$ , that there is at least one element in the set  $\{f_{u_i}(u_1), f_{u_i}(u_2), \dots, f_{u_i}(u_N)\}$ , with  $f_{u_i}(u_k)$  written as a word in  $U$ , of length  $|c_i|_U$ . Hence, all automorphism  $f_t$  of  $\mathcal{H}_{\text{Aut}}$  for which the set  $\{f_t(u_1), f_t(u_2), \dots, f_t(u_N)\}$  has no element of free length  $|c_i|_U$  in  $U$ , was not used by Alice to encrypt  $s_i$  to  $c_i$ . Eve does not know if the element  $c_i$  is completely visibly in  $\hat{C}_{\text{red}}$  or  $\hat{C}_{\text{red}}^{(i)}$ ;
5.  $(i, |c'_i|_U, v, U)$ , with  $c'_i$  an initial segment of  $c_i$  and  $v$  a segment of  $c_i$  ( $c_i \equiv c'_i v \tilde{v}$  with  $\tilde{v}$  word in  $U$  or the empty word), that  $|c_i|_U \geq |c'_i|_U + |v|_U$ . Hence, all automorphisms  $f_t$  of  $\mathcal{H}_{\text{Aut}}$  for which the set  $\{f_t(u_1), f_t(u_2), \dots, f_t(u_N)\}$  has only elements of free length in  $U$  less than  $|c'_i|_U + |v|_U$ , was not used by Alice to encrypt  $s_i$  to  $c_i$ .  
If  $(i, |c'_i|_U, v1, U)$ , then  $|c_i|_U = |c'_i|_U + |v|_U$  and Eve gets the analogous information for the automorphism  $f_{u_i}$  as above in 4. The first  $|c'_i|_U$  elements of  $\hat{C}_{\text{red}}^{(i)}$  are the first  $|c'_i|_U$  elements of  $c_i$  written in  $U$  and after this elements comes the word  $v$  in  $c_i$ , but she does not know, if this word  $v$  is visible in  $\hat{C}_{\text{red}}$  or  $\hat{C}_{\text{red}}^{(i)}$ ;
6.  $(i, 0, c'_i, U)$ , with  $c'_i$  the initial segment of  $c_i$ , that the word  $c_i$  is completely canceled in  $\hat{C}_{\text{red}}$ . She also knows that  $c'_i$  is an initial segment of  $c_i$ , thus  $|c_i|_U \geq |c'_i|_U$ , it is not necessary that  $c'_i = c_i$ . Hence, all automorphisms  $f_t$  of  $\mathcal{H}_{\text{Aut}}$  for which the set  $\{f_t(u_1), f_t(u_2), \dots, f_t(u_N)\}$  has only elements of free length in  $U$  less than  $|c'_i|_U$ , were not used by Alice to encrypt  $s_i$  to  $c_i$ .

In general Eve cannot be sure where  $C_{\text{red}}^{(i)}$  begins (this is equivalent to the beginning of  $c_i$ ) or where  $c_i$  ends. She is also not able to identify all missing letters of  $X^{\pm 1}$  in  $C_{\text{red}}$ , which she needs to get the unreduced word  $C$ .

**Security 8.1.4.** An eavesdropper, Eve, intercepts the reduced ciphertext

$$C_{\text{red}} = x'_1 x'_2 \cdots x'_{L'},$$

with  $x'_i \in X^{\pm 1}$ ,  $1 \leq i \leq L'$ . It is  $C_{\text{red}}$  a word in  $F_U$ .

In general, she is not able to identify the end of  $c_1$  and hence she cannot identify the beginning or end of the other ciphertext units  $c_j$  and she also does not know which elements are canceled in the reduced ciphertext  $C_{\text{red}}$ .

As in Security 8.0.5 to break the system an eavesdropper, Eve, needs to know the set  $U$ . The notations in the following passage are analogous to the notations given in Security 8.0.5.

In an attack of the modification without reduction (beginning of Chapter 8), Eve uses as candidates for the set  $U$  the sets  $V_i = \tilde{C}_{Nred} \cup \{w_1, w_2, \dots, w_l\}$ , with  $w_i$  primitive elements in the ball  $B(F, L_1)$  of the Cayley graph from  $F$  and  $\tilde{C}_{Nred}$  a set of elements which she gets with the help of all ciphertext units  $c_i$ .

Now, in contrast, Eve is neither able to determine the number

$$L_1 := \max\{|c_i|_X \mid c_i \in \tilde{C}\},$$

because she does not know what the ciphertext units  $c_i$  look like and hence she cannot determine the maximum free length of them, nor is she able to generate the set  $\tilde{C}_{Nred}$ , because for this she needs the ciphertext units  $c_i$  unreduced written as a word in  $X$ .

Eve knows  $L' = |C_{red}|_X$  the freely reduced length of the ciphertext. Now, it is likely that she assumes, that the ball  $B(F, L')$  in the Cayley graph for  $F$  contains a basis for  $F_U$ . She searches in the same way for candidates of  $U$  as explained in Security 7.0.8, which is similar as before but gives no guarantee that in all her sets  $V'_i$  is at least one set  $V'$  with  $U_C \subset V'$ .

If Eve gets a candidate  $V'$  for  $U$  she writes the word  $C_{red}$  as a word in  $V'$ , for this she uses the algorithm to solve the constructive membership problem as explained in Theorem 4.3.10. If this is not possible, she knows, that  $V'$  is not the correct set  $U$ , or a set  $V'$ , with  $U_C \subset V'$ . Assume Eve is able to write the ciphertext  $C_{red}$  as a word in her set  $V' = \{v_1, v_2, \dots, v_N\}$ , with  $v_i$  words in  $X$ . Let

$$\hat{C}'_{red} = v'_1 v'_2 \cdots v'_{\ell'},$$

with  $v'_i \in V'^{\pm 1}$ , be the rewritten ciphertext for Eve. It is  $\ell' \leq L'$ , because of Corollary 4.2.12. It is not necessary, that the elements  $v'_i$  are elements in the set  $U^{\pm 1}$ , which was used by Alice and Bob.

The element  $\hat{C}'_{red}$  could give her hints for the used automorphisms by Alice, but these hints are not as good as they were for the cryptosystem without reductions, because Eve does not know which  $v'_j$  is the last letter of  $c_i$ , written as word in  $V'$ , and which  $v'_{j+1}$  is the first letter of  $c_{i+1}$ , written as word in  $V'$ . Maybe there are also letters of  $U$  missing at the end of  $c_i$ , because they are canceled by the initial segment of  $c_{i+1}$ . Thus, it is possible, that Eve assumes  $c_i = v'_i v'_{t+1} \cdots v'_{t+s}$ , but it is  $|c_i|_U > s + 1$ .

It is also likely that there are automorphisms in  $\mathcal{H}_{Aut}$ , for which  $u_i$  has an image which is an initial segment of  $\hat{C}'_{red} = \hat{C}'_{red}$  if  $i = 1$ , or  $\hat{C}'_{red} = c^{i-1} \hat{C}'_{red}^{(i-1)}$  if  $2 \leq i \leq z$ , or  $\hat{C}'_{red} = c_z$  if  $i = z$ . To use the hints for the automorphisms Eve must be able to get the words  $C'^{(i)}_{red}$ , but there are a lot of possibilities for these elements, because Eve does not know where  $c_i$  ends. Furthermore, she does not know if there are cancellations between the ciphertext units and hence if there are letters in the ciphertext  $C_{red}$  missing from which she has no idea. Therefore, she gets more possibilities for the automorphisms than in the version in which the ciphertext is unreduced. Maybe the additional information from Alice, which is sent publicly to Bob, gives her hints (see Remark 8.1.3) but she cannot be sure that there are no other letters missing, which she cannot deduce from the additional information. Hence, if she goes the way described in Security 8.0.5, and in more details in Remark 8.0.7, then it is not likely that, even if she gets an automorphism  $f_i \in \mathcal{H}_{Aut}$  and found in

$$V'_{f_i} = \{f_i(v_1), f_i(v_2), \dots, f_i(v_N)\}$$

a  $j$ , such that  $f_i(v_j)$  is an segment of  $\hat{C}'_{red}$  and thus she assumes that  $c_k = f_i(v_j)$  and  $\hat{C}'_{red} = c_k w$  with  $w$  an terminal segment of  $\hat{C}'_{red}$ , this automorphism  $f_i$  is a used automorphism for encryption and it is not clear that she is on the right way even if in

$$V'_{f_{h(i)}} = \{f_{h(i)}(v_1), f_{h(i)}(v_2), \dots, f_{h(i)}(v_N)\}$$

the next segment of  $C_{red}$ , which is the initial segment of  $\hat{C}'_{red}^{(k+1)}$ , can be found.

## 8.2. Modification with $SL(2, \mathbb{Q})$ for the cryptosystem with $Aut(F_U)$

This modification is analog to the modification with  $SL(2, \mathbb{Q})$  for the cryptosystem with  $Aut(F)$ , see Section 7.2.

Alice and Bob agree also in addition privately on a faithful representation  $\varphi : F \rightarrow SL(2, \mathbb{Q})$  thus they can write the ciphertext units  $c_i$  of the ciphertext  $C$  as matrices in  $SL(2, \mathbb{Q})$  instead as a word in  $X$ . Let

$$\begin{aligned} \varphi : F &\rightarrow SL(2, \mathbb{Q}) \\ x_i &\mapsto M_i \end{aligned}$$

be this faithful representation of  $F$  into  $SL(2, \mathbb{Q})$ .

One way for Alice and Bob to generate the matrices  $M_i$ ,  $1 \leq i \leq q$ , is given in Remark 7.2.2. In addition to the situation in Section 7.2, the ciphertext  $C$  can now also be seen as a word in  $U$  and therefore the ciphertext  $C'$  is also a word in  $U_\varphi$ . Thus, Alice and Bob must take care, that the set  $U_\varphi = \{\varphi(u_1), \varphi(u_2), \dots, \varphi(u_N)\}$ , with  $\varphi(u_i) \in SL(2, \mathbb{Q})$  is not a generating set of a subgroup in  $SL(2, \mathbb{Q})$  for which the membership problem is decidable, therefore,  $|N| \geq 3$  and  $\langle U_\varphi \rangle$  is not a subgroup of  $SL(2, \mathbb{Z})$ .

The encryption is realizable with a table (like Table 8.1 (page 190)) if the representation  $\varphi$  is applied to the elements  $f_{u_i}(u_j)$  in this table, see Table 8.7 (page 205).

Table 8.7.: Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  corresponding to ciphertext alphabet  $U_{\varphi(f_{u_i})}$  depending on the automorphisms  $f_{u_i}$  and the faithful representation  $\varphi$

	$U_{\varphi(f_{u_1})}$	$U_{\varphi(f_{u_2})}$	$\dots$	$U_{\varphi(f_{u_z})}$
$a_1$	$\varphi(f_{u_1}(u_1))$	$\varphi(f_{u_2}(u_1))$	$\dots$	$\varphi(f_{u_z}(u_1))$
$a_2$	$\varphi(f_{u_1}(u_2))$	$\varphi(f_{u_2}(u_2))$	$\dots$	$\varphi(f_{u_z}(u_2))$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_N$	$\varphi(f_{u_1}(u_N))$	$\varphi(f_{u_2}(u_N))$	$\dots$	$\varphi(f_{u_z}(u_N))$

The encryption for Bob is not possible with the use of the solution of the constructive Membership problem for the ciphertext  $C'$ , a sequence of matrices, the set  $U_\varphi$  and the used automorphisms  $f_{u_i}$ , because Alice and Bob take care that there is no algorithm known to solve this problem efficiently.

**Security 8.2.1.** As in Security 7.2.1 the additional security certification is, that there is no algorithm known to solve the membership problem (see Problem 4.3.8) for free (discrete) subgroups of  $SL(2, \mathbb{Q})$  which are of rank greater than or equal to 2 and not subgroups of  $SL(2, \mathbb{Z})$ , even if Eve makes the correct guess for the set  $M = \{M_1, M_2, \dots, M_q\}$ , with  $M_i = \varphi(x_i)$ , used by Alice and Bob, or the set  $U_\varphi = \{\varphi(u_1), \varphi(u_2), \dots, \varphi(u_N)\}$ , with  $\varphi(u_i) \in SL(2, \mathbb{Q})$ , which

could also be used for encryption, see above. Thus, Eve is not able to generate a situation as in Security 8.0.5, hence it is very unlikely that Eve is able to decrypt the message correctly. Even if Eve does a brute force search through the set  $\mathcal{H}_{Aut}$  with her candidate  $U'_\varphi$  for the set  $U_\varphi$  instead of the abstract set  $U$  and finds with her candidate  $U'_\varphi$  and an automorphism of  $\mathcal{H}_{Aut}$  one matrix  $W_i$  of the ciphertext  $C'$  it could be that she used a different  $U'_\varphi$  and different automorphism of  $\mathcal{H}_{Aut}$  than Alice and hence encrypts this ciphertext unit  $W_i$  not correctly.

**Remark 8.2.2.** If Alice and Bob use this variation, then they can also let the set  $X$  private. In the previous version the ciphertext  $C$  is a sequence of words in  $X$  hence if Alice looks at  $C$  it is very likely that she gets all elements in  $X$ , hence the set  $X$  is considered as public. In addition the automorphisms of  $\mathcal{F}_{Aut}$  give all elements in  $X$ . Now, the ciphertext  $C'$  is a sequence of matrices in  $SL(2, \mathbb{Q})$  and Alice and Bob can choose  $X$  privately. This makes a correct guess from Eve for the set  $M$  more difficult, because Eve does not know the cardinality of  $M$ , which is  $q = |X|$ , because know  $X$  is private. The automorphisms in  $\mathcal{H}_{Aut}$  give no hint, because they are abstract given on a set with  $N$  elements and in general it is  $N \neq |X|$ .

### 8.3. Modification with Hilbert's Tenth Problem for the cryptosystem with $Aut(F_U)$

The same way how Alice and Bob use Hilbert's Tenth Problem for the cryptosystem with  $Aut(F)$  in Section 7.3, they can use it for the cryptosystem with  $Aut(F_U)$ . Alice writes the ciphertext  $C$ , which she gets by the cryptosystem with  $Aut(F_U)$  to a sequence  $C_{Hilbert}$  of matrices in  $GL(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  and  $n \geq 2$ , as explained in Section 7.3, because in both cryptosystems (with  $Aut(F)$  and  $Aut(F_U)$ ) the ciphertext is a sequence of words in  $X$ , with  $X = \{x_1, x_2, \dots, x_q\}$  and  $F = \langle X \mid \quad \rangle$ . Therefore, the decryption from  $C_{Hilbert}$  to  $C$  is for Bob the same as in Section 7.3 explained.

**Remark 8.3.1.** The version of the cryptosystem of  $Aut(F_U)$  in which the ciphertext is only one reduced word in  $X$  (Section 8.1) can also be improved with the explained procedure which uses the negative solution of Hilbert's Tenth Problem, analogous to Remark 7.3.8. Then the ciphertext is only a matrix  $M_{Hilbert}$  in  $GL(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  and  $n \geq 2$ .

**Security 8.3.2.** The security certification depends, in addition to Security 8.0.5, (for Remark 8.3.1 in addition to Security 8.1.4) on the unsolvability of Hilbert's Tenth Problem. Y. Matiyasevich proved in [Mat70] finally that there is no general algorithm which determines whether or not an integral polynomial in any number of variables has a zero. Therefore, for Eve, who sees just matrices in  $GL(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  and  $n \geq 2$ , it is hard to find the private key  $D$  of Alice and Bob.

We get one more variation of this cryptosystem, which is only possible if we use in addition the negative solution of Hilbert's Tenth Problem. Let

$$C = c_1 c_2 \cdots c_z$$

be the ciphertext generated as explained in the beginning of Chapter 8. The ciphertext units  $c_i$  are words in  $U$ , with  $U = \{u_1, u_2, \dots, u_N\}$  a Nielsen reduced set with elements  $u_i$ , which are words in  $X$ , hence the  $c_i$  are also words in  $X$ . In the above systems with  $Aut(F)$  and  $Aut(F_U)$ , the ciphertext  $C$  was send as a sequence of words in  $X$ . Now, we consider the ciphertext units  $c_i$  as words in the abstract set  $U$ .

To make use of Hilbert's Tenth Problem, the augmentation here (as in Section 7.3) is given by evaluating a point  $D = \{d_1, d_2, \dots, d_n\} \in \mathbb{Z}^n$ , such that

$$\epsilon^*(A) = \begin{pmatrix} p_1(D) & p_2(D) \\ p_3(D) & p_4(D) \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = a$$

and

$$\epsilon^*(B) = \begin{pmatrix} p_5(D) & p_6(D) \\ p_7(D) & p_8(D) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} = b.$$

The **public** knowledge for this modification extends to the augmented ring  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ , with  $n \in \mathbb{N} \setminus \{1\}$ .

The **private** information for Alice and Bob extends to a point  $D \in \mathbb{Z}^n$ , the set  $\{a, b\}$  and the Nielsen reduced set  $V = \{v_1, v_2, \dots, v_N\}$ , with  $v_i$  abstract words in  $\{a, b\}$ .

In this modification the elements in the abstract set  $U$  are identified with words in the set  $\{a, b\}$  instead as words in  $X$ . Thus, instead of a free subgroup  $J_W = \langle W \mid \quad \rangle$  with rank  $q$  of  $J = \langle a, b \mid \quad \rangle$  (see Section 7.3) Alice and Bob use a free subgroup  $J_V = \langle V \mid \quad \rangle$  with rank  $N$ ,  $N = |U|$  the number of letters in the used plaintext alphabet  $A$ . Therefore, let  $V = \{v_1, v_2, \dots, v_N\}$  be a Nielsen reduced set with  $v_i$ ,  $1 \leq i \leq N$ , abstract words in  $\{a, b\}$ .

Alice and Bob identify  $u_i \in U$  with  $v_i \in V$ , for all  $i = 1, 2, \dots, N$ , and thus Alice writes the ciphertext (generated as explained in Chapter 8)

$$C = c_1 c_2 \cdots c_z,$$

with  $c_i$  abstract words in  $U$ , as

$$C' = c'_1 c'_2 \cdots c'_z,$$

with  $c'_i$  abstract words in  $V$ .

Next, Alice writes  $C'$  as a sequence of words in  $\{A_j, B_j\}$ , that is,  $C'_{Hilbert}$  and means, instead of  $a$  she writes  $A_j$  and instead of  $b$  she writes  $B_j$ . It is

$$A_j = \begin{pmatrix} p_{1_j} & p_{2_j} \\ p_{3_j} & p_{4_j} \end{pmatrix} \quad \text{and} \quad B_j = \begin{pmatrix} p_{5_j} & p_{6_j} \\ p_{7_j} & p_{8_j} \end{pmatrix},$$

with  $p_{1_j}, p_{2_j}, \dots, p_{8_j} \in \mathbb{Z}[y_1, y_2, \dots, y_n]$ ,  $n \geq 2$ , her 8 ephemeral polynomials, which generate the matrices  $A_j$  and  $B_j$  in  $\text{GL}(2, R)$  with the property

$$\epsilon^*(A_j) = a \quad \text{and} \quad \epsilon^*(B_j) = b.$$

Thus,

$$C'_{Hilbert} = \hat{c}_1 \wr \hat{c}_2 \wr \cdots \wr \hat{c}_z$$

is the ciphertext  $C'$  written as a sequence of matrices in  $\text{GL}(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  and  $n \geq 2$ . Hence, Alice sends each  $\hat{c}_i$ ,  $1 \leq i \leq z$ , as one matrix in  $\text{GL}(2, R)$  to Bob.

For **decryption** Bob uses the augmentation, which is given by evaluating  $p_{i_j}$ ,  $1 \leq i \leq 8$ , at the private point  $D$ , such that  $\epsilon^*(A_j) = a$  and  $\epsilon^*(B_j) = b$ . With this point  $D$  he is able to generate from  $C'_{Hilbert}$  the ciphertext version

$$\hat{C}' = \epsilon^*(\hat{c}_1) \wr \epsilon^*(\hat{c}_2) \wr \cdots \wr \epsilon^*(\hat{c}_z) = \hat{c}'_1 \wr \hat{c}'_2 \wr \cdots \wr \hat{c}'_z,$$

with  $c'_i$ ,  $1 \leq i \leq z$ , matrices in  $SL(2, \mathbb{Z})$ , which are words in  $\{a, b\}$ . With an algorithm (use for example the method described in Remark 4.3.12) to write the matrix  $c'_i$  as an abstract word  $c'_i$  in  $\{a, b\}$  he gets

$$C' = c'_1 \wr c'_2 \wr \cdots \wr c'_z.$$

Since Alice and Bob choose a Nielsen reduced set  $V = \{v_1, v_2, \dots, v_q\}$ ,  $v_j$  abstract words in  $\{a, b\}$ , Bob is now able to write each  $c'_i$ ,  $1 \leq i \leq z$ , as an abstract word in  $V$ , see Theorem 4.3.10 and Remark 4.3.11, and gets

$$C' = c_1 \wr c_2 \wr \cdots \wr c_z,$$

with  $c_i$ ,  $1 \leq i \leq z$ , words in  $V$ .

Bob writes the ciphertext  $C'$  with the one-to-one correspondence between  $U$  and  $V$  to the ciphertext  $C$ , in which the ciphertext units are written as words in  $U$ . Thus, the decryption is then the same as explained in **Encryption and Decryption Procedure 3.1.b** in the beginning of Chapter 8.

**Remark 8.3.3.** This modification is only possible with the use of the negative solution of Hilbert's Tenth Problem. Without writing the elements  $c'_i$  as matrices in  $GL(2, R)$  with the integral polynomial ring  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ ,  $n \geq 2$ , an eavesdropper intercepts ciphertext units as abstract words in  $U$  or  $V$ , respectively. Alice and Bob use the one-to-one correspondence with  $u_i \mapsto v_i$  for  $u_i \in U$  and  $v_i \in V$ . This gives hints for the used automorphism of Alice in  $\mathcal{H}_{Aut}$  for encryption, because now the ciphertext units are images of elements in  $U$  or  $V$ , respectively, and Eve knows that Alice used an automorphism for which the ciphertext unit  $c_i$  is such an image. The abstract automorphisms  $\mathcal{H}_{Aut} \subset Aut(H)$  on  $U$ , see Chapter 8, can be also seen as abstract automorphisms on  $V$  and  $H' = \langle V \mid \ \rangle$  instead on  $H = \langle U \mid \ \rangle$  (this is possible because  $|V| = |U|$  and hence  $H$  is isomorphic to  $H'$ , see Theorem 4.3.7), thus  $\mathcal{H}_{Aut} = \mathcal{H}'_{Aut}$  and  $Aut(H) = Aut(H')$ .

**Security 8.3.4.** The security certification depends on the unsolvability of Hilbert's Tenth Problem. Y. Matiyasevich proved in [Mat70] finally that there is no general algorithm which determines whether or not an integral polynomial in any number of variables has a zero. Therefore, for Eve, who sees just matrices in  $GL(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$ , it is hard to find the private key  $D$  of Alice and Bob. In addition the security is improved by the fact, that for each encryption Alice and Bob can take privately ephemeral matrices in  $GL(2, R)$  with the property that the common private point  $D \in \mathbb{Z}^n$  generates the correct matrices in  $PSL(2, \mathbb{Z})$ . This gives randomness to ciphertexts, which complicates attacks for Eve.

## 8.4. Chosen plaintext attacks on the cryptosystem with $Aut(F_U)$

In a chosen plaintext attack (see Section 1.1 or for instance also [BFKR15, Section 3.1]) Eve gives a blackbox, which does the encryption procedure, plaintexts of her choice and gets the corresponding ciphertexts. Assume Eve gives the blackbox a plaintext which obtains only one letter of the alphabet  $A$ , it is

$$\begin{aligned} S &= s_1 s_2 \cdots s_r \\ &= \underbrace{a_i a_i \cdots a_i}_{r \text{ times}}, \end{aligned}$$



with  $a_i \in A$ . She gets the ciphertext

$$\begin{aligned} C &= f_\alpha(a_i) f_{h(\alpha)}(a_i) f_{h^2(\alpha)} \cdots (a_i) f_{h^{r-1}(\alpha)}(a_i) \\ &= c_1 c_2 \cdots c_r, \end{aligned}$$

with  $\alpha$  the starting seed to generate with the linear congruence generator  $h$  the used automorphisms  $f_i \in \mathcal{H}_{Aut}$  for encryption.

The ciphertext units  $c_i$  are words in  $U$ , which are given as words in  $X$ , because  $U = \{u_1, u_2, \dots, u_N\}$  and  $u_i$  words in  $X$ .

In general, the ciphertext units do not have a similar structure as it could be in the case for a chosen plaintext attack of the cryptosystem with  $Aut(F)$ , see Section 7.4.

Eve gets words  $f_{h^{j-1}(\alpha)}(a_i) = c_j$ ,  $1 \leq j \leq r$ , in  $X$  which are differently written in  $U$  and it is also possible that  $u_i^\epsilon$ ,  $\epsilon \in \{1, -1\}$ , is not a letter in the word  $c_j$ .

Thus, a chosen plaintext attack gives no additional hints for the set  $U$ , than the look at the ciphertext, which is a ciphertext only attack, see Security 8.0.5 and especially Remark 8.0.6.

**Example 8.4.1.** Eve gives the blackbox the plaintext

$$S = \text{LLLL.}$$

We assume that the blackbox used the same public and private parameters as in Example 8.0.4. The starting seed for the linear congruence generator  $h$  is  $u_1 = \bar{\alpha} = \overline{23442}$  and the used automorphisms  $f_{u_1}$ ,  $f_{u_2}$ ,  $f_{u_3}$  and  $f_{u_4}$ , for encryption are computed as above in Example 8.0.4, hence the ciphertext is

$$\begin{aligned} C &= f_{u_1}(\text{L}) f_{u_2}(\text{L}) f_{u_3}(\text{L}) f_{u_4}(\text{L}) \\ &= c_1 c_2 c_3 c_4 \\ &= xyzx^3 y^2 zy^{-2} x^2 \wr (xz^{-1})^2 y^{-1} x^{-1} yzy^{-2} x^2 \wr xyzzyz^{-1} y^{-2} x^{-2} z^{-1} xy^3 z^{-2} \wr y^{-1} xzyz. \end{aligned}$$

There is no structure for  $u_1 \hat{=} a_1 = \text{L}$  identifiable. The following is a consideration, which Eve could also take at a ciphertext only attack, see Remark 8.0.6. Eve could assume that one  $u_i$  is of the form or has a segment or is a segment of

$$zyy^{-2}x^2$$

because this segment is a terminal segment of  $c_1$  and  $c_2$ .

It is also likely, that she assumes, that one  $u_j$  is of the form or has a segment or is a segment of

$$xyz$$

because this segment is an initial segment of  $c_1$  and  $c_3$ .

Remember, that in Example 8.0.4, Alice and Bob agreed on

$$\begin{aligned} u_1 &:= xyz, & u_2 &:= yzy^{-1}, & u_3 &:= x^{-1}zx^{-1}, & u_4 &:= y^{-1}x^2, \\ u_5 &:= z^{-1}xyx, & u_6 &:= z^{-1}yx^{-1}, & u_7 &:= x^3y, & u_8 &:= y^3z^{-2}. \end{aligned}$$

Therefore, she takes correct thought for  $u_1 = xyz$  and  $u_4 = y^{-1}x^2$ , this enables a more selective brute force search through the Cayley graph for the set  $U$ , see Security 8.0.5.

If we are in the situation (Section 8.1) in which the ciphertext  $C_{red}$  is a reduced word and the beginning and end of a ciphertext unit  $c_i$  is not marked it is more difficult to get information about a blackbox if words are used as inputs. If only one letter at the time is an input for the

blackbox, then they get the same information as in the situation when the ciphertext units are identifiable in the unreduced ciphertext.

If we are in the situation (Section 8.2) in which the ciphertext  $C'$  is a matrix in  $\text{SL}(2, \mathbb{Q})$  this attack gives no information about the way the elements in the set  $U$  look like, because an eavesdropper, Eve, sees just matrices and she does not know which matrices are multiplied to get the ciphertext matrices  $c_i$ .

If we are in the situation (Section 8.3) in which the ciphertext  $C'_{\text{Hilbert}}$  is a sequence of matrices in  $\text{GL}(2, R)$ , with  $R = [y_1, y_2, \dots, y_n]$ , this attack gives no hint for the elements in  $U$ . As in the situation in which the ciphertext is a sequence of matrices in  $\text{SL}(2, \mathbb{Q})$  an eavesdropper sees just matrices and she does not know which matrices are multiplied to get the ciphertext matrices  $\hat{c}_i$ .

### Conclusion concerning chosen plaintext attacks

If the ciphertext is given as a matrix, the system is secure against chosen plaintext attacks. If the ciphertext is a word in  $X$  it could be possible that an eavesdropper can get hints for the elements in  $U$  and hence the search for the primitive elements in the Cayley graph could be performed in a more selective measure, but these hints can also be seen in a ciphertext only attack. Hence, this is not an information which only appears at a chosen plaintext attack.

## 8.5. Chosen ciphertext attacks on the cryptosystem with $\text{Aut}(F_U)$

In a chosen ciphertext attack (see Section 1.1 or for instance also [BFKR15, Section 3.1]) Eve chooses ciphertexts and sends these to a blackbox, she then gets the corresponding plaintexts back.

For example Eve gets to some parts of a given ciphertext the corresponding plaintext. These parts can be chosen by her.

Let

$$C = c_1 c_2 \cdots c_z$$

be the ciphertext generated as explained in Chapter 8.

Let

$$c_j = f_{u_j}(u_k)$$

be the ciphertext unit for which Eve knows the plaintext  $s_j = a_k$ . She gets no more information about the set  $U$ , than she also knows from a ciphertext only attack.

She knows the free length  $|c_j|_X$  and thus  $|c_j|_U \leq |c_j|_X$  because of Corollary 4.2.12.

Eve is able, as before (see Security 8.0.5 more precisely Remark 8.0.7), to reduce the set  $\mathcal{H}_{\text{Aut}}$ , which gives abstract automorphisms of  $F_U = \langle U \mid \rangle$  for an abstract set  $U = \{u_1, u_2, \dots, u_N\}$ , for the automorphism which Alice used to encrypt  $s_j = a_k$  with her explicit set  $U$ , in which the elements are words in  $X$ . Only automorphisms  $f_\ell \in \mathcal{H}_{\text{Aut}}$  for which  $|f_\ell(u_k)|_U \leq |c_j|_X$  is true are candidates for the used automorphism  $f_{u_j}$  by Alice.

Assume Eve gets a candidate  $V' = \{v_1, v_2, \dots, v_N\}$ , with  $v_i$  words in  $X$ , for Alice set  $U$ . She writes the element  $c_j$  as

$$c_j = v_{j_1} v_{j_2} \cdots v_{j_l}, \tag{8.1}$$

with  $v_{j_i} \in V'^{\pm 1}$  for  $1 \leq i \leq l$  and  $l \leq |c_j|_X$  (with the algorithm given in Theorem 4.3.10). If  $l > |c_j|_X$  the set  $V'$  is wrong.

This is a hint for the used automorphism by Alice. Therefore, Eve searches in the set  $\mathcal{H}_{\text{Aut}}$  of automorphisms on  $U$  for such an automorphism, which applies the element  $u_k$  to an element  $u_{j_1}^{\epsilon_1} u_{j_2}^{\epsilon_2} \cdots u_{j_l}^{\epsilon_l}$  with  $j_a \in \{1, 2, \dots, N\}$  for  $1 \leq a \leq l$  and  $\epsilon_b \in \{1, -1\}$  for  $1 \leq b \leq l$ . If  $v_{j_p} = v_{j_q}$ ,  $p \neq q$ , in Eve's equation (8.1), then  $u_{j_p}^{\epsilon_p} = u_{j_q}^{\epsilon_q}$ , and  $v_{j_p}$  corresponds to  $u_{j_p}^{\epsilon_p}$ . Now, she only searches for this explicit  $u_k$  and not for all  $u_r$  with  $1 \leq r \leq N$ . Thus, this search could be performed in a more selective measure as in Security 8.0.5, more precisely Remark 8.0.7.

Assume we are in the situation in which the ciphertext  $C_{red}$  is a reduced word in  $X$  and the beginning and end of each ciphertext unit  $c_i$  is not marked (see Section 8.1).

With a chosen ciphertext attack Eve gets information how the word  $C_{red}$  is assigned partly to the plaintext units. In general, Eve is not able to write  $C_{red}$  as  $C$ , that means, she is not able to identify for each ciphertext unit  $c_i$  the end and the beginning and she also does not know all cancellations between  $c_i$  and  $c_{i+1}$ . Hence, to find the set  $U$ , she gets no additional hints for  $U$  than in a ciphertext only attack (see Security 8.1.4) on  $C_{red}$ . To get candidates  $V'$  for  $U$  she acts as described in Security 8.1.4.

There could appear different cases for her information which she gets from a chosen ciphertext attack (these are similar to the cases for the cryptosystem with  $\text{Aut}(F)$ , see Section 7.5):

1. If there are no cancellations for  $c_j$  and Alice knows which part of  $C_{red}$  corresponds to  $s_j$  then she knows  $c_j$  and knows where it ends and begins and hence where  $c_{j+1}$  begins and  $c_{j-1}$  ends. In general she does not know which number of  $\{1, 2, \dots, z\}$  is  $j$ . She now could act similar as in the unreduced ciphertext case above if she gets a candidate for  $U$ .
2. If there are cancellations for  $c_j$  it is not sure that Eve knows that. Maybe it is not necessary for Alice to give Bob such an additional information and hence Eve does not know if there are cancellations or not. Let

$$C_{red} \equiv w_1 \tilde{c}_j w_2$$

be the ciphertext with  $w_1, w_2$  words in  $X$  and  $c_j \equiv c_{j_1} \tilde{c}_j c_{j_2}$  with  $c_{j_1}, c_{j_2}$  words in  $X$  or the empty word. In a chosen ciphertext attack Eve gets the information

$$\tilde{c}_j \xrightarrow{\text{is decrypted to}} s_j = a_k (\hat{=} u_k),$$

for a  $k \in \{1, 2, \dots, N\}$ . Thus, her information is  $c_j \equiv c_{j_1} \tilde{c}_j c_{j_2}$  with  $c_{j_1}, c_{j_2}$  words in  $X$  or the empty word, and she does not know what  $c_{j_1}$  and  $c_{j_2}$  look like. In general she does not know which number of  $1, 2, \dots, z$  is  $j$ .

If Eve gets a candidate  $V'$  for  $U$  she writes the word  $C_{red}$  as a word in  $V'$ , for this she uses the algorithm to solve the constructive membership problem as explained in Theorem 4.3.10. If this is not possible, she knows, that  $V'$  is not the correct set  $U$ , or a set  $V'$ , with  $U_C \subset V'$ . Assume Eve is able to write the ciphertext  $C_{red}$  as a word in her set  $V' = \{v_1, v_2, \dots, v_N\}$ , with  $v_i$  words in  $X$ . Let

$$\hat{C}'_{red} = v'_1 v'_2 \cdots v'_{\ell'},$$

with  $v'_i \in V'^{\pm 1}$ , be the rewritten ciphertext for Eve. It is  $\ell' \leq |C_{red}|_X$ , because of Corollary 4.2.12. She then also knows which elements of  $V'$  correspond to the element  $\tilde{c}_j$ , that means she knows

$$\tilde{c}_j = v'_t v'_{t+1} \cdots v'_{t+d}, \tag{8.2}$$

with  $v'_i \in V'^{\pm 1}$ ,  $t \in \{1, 2, \dots, \ell' - d\}$  and  $d+1 = |\tilde{c}_j|_{V'} \leq |c_j|_X$ , because of Corollary 4.2.12. This is a hint for the used automorphism by Alice. Therefore, Eve searches in the set  $\mathcal{H}_{\text{Aut}}$  of automorphisms on  $U$  for such an automorphism, which applies the element  $u_k$  to an element  $w_1 u_{j_1}^{\epsilon_1} u_{j_2}^{\epsilon_2} \cdots u_{j_{d+1}}^{\epsilon_{d+1}} w_2$ , with  $w_1, w_2$  words in  $U$  or the empty word and  $j_a \in \{1, 2, \dots, N\}$  for  $1 \leq a \leq d+1$  and  $\epsilon_b \in \{1, -1\}$  for  $1 \leq b \leq d+1$ . If  $v'_p = v'_q$ ,  $p \neq q$ , in Eve's equation (8.2), then  $u_{j_p}^{\epsilon_p} = u_{j_q}^{\epsilon_q}$ , and  $v'_p$  corresponds to  $u_{j_p}^{\epsilon_p}$ . Now, she only searches for this explicit  $u_k$  and not for all  $u_r$  with  $1 \leq r \leq N$ . Hence, the search for the possible automorphisms of Alice in  $\mathcal{H}_{\text{Aut}}$  could be performed in a more selective measure as in Security 8.0.5, more precisely Remark 8.0.7, described.

Even if Eve finds such an automorphism she cannot be sure that it is the automorphism of Alice or her set  $V'$  is correct, see also Security 8.1.4.

It is difficult for Eve to find the used set  $U$  and the automorphisms which were used by Alice and Bob.

3. If the ciphertext is a sequence of different ciphertexts  $C_{\text{red}_i}$ ,  $1 \leq i \leq z'$ , given as words in  $X$ , and she gets a plaintext, for example for one  $C_{\text{red}_j}$ , she then knows of how many letters in  $U$ , and hence in  $A$  (the plaintext alphabet), the message  $C_{\text{red}_j} = c_{j_1} c_{j_2} \cdots c_{j_{z_j}}$  is written, that means, she knows  $|C_{\text{red}_j}|_U = z_j$ . In general, she does not know where each ciphertext unit  $c_{j_k}$ ,  $1 \leq k \leq z_j$ , begins or ends. Or if there are cancellations between the ciphertext units  $c_{j_k}$  and  $c_{j_{k+1}}$ .

This gives no more information for the set  $U$  as in the cases above.

Assume we are in the situation in which the ciphertext  $C'$  is a sequence of matrices in  $\text{SL}(2, \mathbb{Q})$ , see Section 8.2. Let

$$C' = W_1 W_2 \cdots W_z$$

be the ciphertext with  $W_i \in \text{SL}(2, \mathbb{Q})$ ,  $1 \leq i \leq z$ . With the chosen ciphertext attack, Eve gets for example the information

$$W_j \xrightarrow{\text{is decrypted to}} s_j = a_k (\hat{=} u_k),$$

with  $1 \leq j \leq z - 1$  and  $k \in \{1, 2, \dots, N\}$ . There is no hint for Eve for the used set  $M$ , to get the faithful representation  $\varphi$ , or the set  $U$ , used by Alice and Bob, and hence for the set  $U_\varphi$ . Only the brute force search described in Security 8.2.1 could be performed in a more selective measure, that means, Eve makes a guess  $U'_\varphi = \{\tilde{M}_1, \tilde{M}_2, \dots, \tilde{M}_N\}$ , with  $\tilde{M}_j \in \text{SL}(2, \mathbb{Q})$ , for the set  $U_\varphi$  of Alice and Bob. Eve tests if

$$f_\ell(\tilde{M}_k) = W_j,$$

for  $f_\ell \in \mathcal{H}_{\text{Aut}}$  (the automorphisms in  $\mathcal{H}_{\text{Aut}}$  can be applied on  $\langle U'_\varphi \rangle$  because  $|U'_\varphi| = |U|$  and therefore the free group  $F_U = \langle U \mid \ \rangle$  is isomorphic to  $F_{U'_\varphi} = \langle U'_\varphi \mid \ \rangle$ ), instead if

$$W_j \in f_\ell(U'_\varphi).$$

Even if she found a set  $U'_\varphi$  and an automorphism  $f_\ell \in \mathcal{H}_{\text{Aut}}$ , such that

$$f_\ell(\tilde{M}_k) = W_j$$

she cannot be sure that these were used by Alice and Bob. There are a lot of candidates for  $U'_\varphi$ , that means sets in  $\text{SL}(2, \mathbb{Q})$  with  $N$  elements, such that

$$f_\ell(\tilde{M}_k) = W_j$$

for some  $f_\ell \in \mathcal{H}_{\text{Aut}}$ .

Assume we are in the situation in which the ciphertext  $C_{\text{Hilbert}}$  is a sequence of matrices in  $\text{GL}(2, R)$ , with  $R = \mathbb{Z}[y_1, y_2, \dots, y_n]$  and  $n \geq 2$ , see Section 8.3. This situation is analogous to the modification in which the ciphertext is a sequence of matrices in  $\text{SL}(2, \mathbb{Q})$ . Eve gets neither a hint for  $U$  nor  $A_j$  and  $B_j$  nor  $D$  nor  $W$ .

Conclusion concerning chosen ciphertext attacks

This cryptosystem is secure against chosen ciphertext attacks. An attacker gets no additional hints for the set  $U$  than he gets with a ciphertext only attack.



## Chapter 9

# Private key cryptosystem which uses automorphisms on plaintext sequences (Protocol 10)

This chapter introduces **Protocol 10**, a symmetric key cryptosystem. It is based on combinatorial group theory, uses automorphisms of finitely generated free groups, Nielsen reduced sets and a faithful representation of a finitely generated free group into  $SL(2, \mathbb{Q})$ . The automorphisms are out of a common set  $\mathcal{G}_{Aut} \subset Aut(G)$  (with  $G$  an abstract free group of finite rank). For decryption Bob needs to know which automorphisms of  $\mathcal{G}_{Aut}$  were used for the encryption procedure by Alice. For this choice of elements in  $\mathcal{G}_{Aut}$  regulations are needed. Therefore, Alice and Bob make use of a linear congruence generator with maximal periodic length as for **Protocol 8** and **Protocol 9**. Hence, for linear congruence generators see Chapter 7.

The main difference of the cryptographic protocol in this chapter to **Protocol 8** and **Protocol 9** is, that it uses automorphisms on plaintext sequences instead of automorphisms on  $F$  or  $F_U$ , respectively. Moreover, **Protocol 10** contains special random matrices, which generate randomness for the ciphertext and work as ephemeral keys in the encryption procedure.

We first describe **Protocol 10** in a restricted version to explain the idea. This is then generalized. A variation and an example are given. The chapter closes with a closer look at chosen ciphertext and chosen plaintext attacks.

Now, we introduce **Protocol 10**.

### Public Parameters

First of all Alice and Bob agree on public parameters.

1. A plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  with  $N \geq 2$ .
2. An abstract free group  $G = \langle Y \mid \quad \rangle$  with  $rank(G) = 5$  and a free generating set  $Y = \{y_1, y_2, \dots, y_5\}$ .
3. A subset  $\mathcal{G}_{Aut} := \{g_0, g_1, \dots, g_{2^{128}-1}\} \subset Aut(G)$  of automorphisms of  $G$ . It is  $g_i : G \rightarrow G$  and the  $g_i$ ,  $i = 0, 1, \dots, 2^{128} - 1$ , pairwise different, are generated with the help of the 0-1-sequence (of different length) and random numbers as described in Section 4.4. The set  $\mathcal{G}_{Aut}$  is part of the key space.
4. They agree on a linear congruence generator  $h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}}$  with a maximal period length.

### Private Parameters

Afterwards they agree on their common private parameters.

1. A finitely generated free group  $F = \langle X \mid \ \rangle$  with free generating set  $X = \{x_1, x_2, \dots, x_q\}$ , with  $q \geq 2$ .
2. A free subgroup  $F_U = \langle U \mid \ \rangle$  with rank  $2N$  of the free group  $F$  and the free generating set  $U = \{u_1, u_2, \dots, u_N, u_{N+1}, \dots, u_{2N}\}$ , with  $u_i$  freely reduced words in  $X$ , is chosen. The set  $U$  is a minimal Nielsen reduced set (with respect to a lexicographical order, see for instance Example 4.2.8). Such systems  $U$  are easily to construct using Theorem 4.2.13 and Lemma 4.2.15 (see also [CgRR08] and [LS77]). It is  $\mathcal{U}_{Nred}$  the set of all minimal Nielsen reduced sets with  $2N$  elements in  $F$ , which is part of the key space.
3. They agree on a faithful representation

$$\begin{aligned} \varphi : F &\rightarrow \mathrm{SL}(2, \mathbb{Q}) \\ x_i &\mapsto M_i. \end{aligned}$$

The set  $M = \{M_1, M_2, \dots, M_q\}$ , with  $M_i = \varphi(x_i)$ , is a free generating set for a free subgroup  $H = \langle M \mid \ \rangle$  of  $\mathrm{SL}(2, \mathbb{Q})$ . The groups  $F$  and  $H$  are isomorphic.

4. With the representation  $\varphi$  they calculate the free group  $F_{U'} = \langle U' \mid \ \rangle$ . The free generating set is  $U' = \{V_1, V_2, \dots, V_{2N}\}$ , with  $V_i = \varphi(u_i)$  and  $1 \leq i \leq 2N$ . Hence, the group elements are matrices in  $\mathrm{SL}(2, \mathbb{Q})$ . The group  $F_U$  is isomorphic to the group  $F_{U'}$ .
5. They agree on the assignment

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N}.$$

6. Alice and Bob agree on an automorphism  $g_{\bar{\alpha}} \in \mathcal{G}_{Aut}$ , hence  $\alpha$  is the common secret starting point  $\alpha \in \{0, 1, \dots, 2^{128} - 1\}$ , with  $y_1 = \bar{\alpha} \in \mathbb{Z}_{128}$  for the linear congruence generator  $h$ . With this  $\alpha$  they are able to generate the sequence of automorphisms of the set  $\mathcal{G}_{Aut}$ , which is  $y_1 = \bar{\alpha} \in \mathbb{Z}_{2^{128}}$ ,  $y_2 = h(y_1) \in \mathbb{Z}_{2^{128}}$ ,  $\dots$ ,  $y_\beta = h(y_{\beta-1}) \in \mathbb{Z}_{2^{128}}$ . The sequence  $g_{y_1}, g_{y_2}, \dots, g_{y_\beta}$  of automorphisms of the set  $\mathcal{G}_{Aut}$  is used for encryption and to generate the automorphisms for decryption of a plaintext with  $z$  plaintext letters, it is  $\beta = \lceil \frac{z}{4} \rceil$  (for  $r \in \mathbb{R}$  it is  $\lceil r \rceil := \min\{x \in \mathbb{Z} \mid x \geq r\}$ ).

**Key space:**  $F = \langle X \mid \ \rangle$ ,  $X = \{x_1, x_2, \dots, x_q\}$  with  $q \geq 2$ ; the set  $\mathcal{U}_{Nred}$  of all minimal (with respect to a lexicographical order) Nielsen reduced subsets of  $F$  with  $2N$  elements. The set  $\mathcal{G}_{Aut} \subset \mathrm{Aut}(G)$  of  $2^{128}$  randomly chosen automorphisms of  $G$ .

### Private Key Cryptosystem

Now, we explain the private key cryptosystem and look carefully at the steps for Alice and Bob.

**Public knowledge:** Plaintext alphabet  $A = \{a_1, a_2, \dots, a_N\}$  with  $N \geq 2$ ,  $G = \langle Y \mid \ \rangle$ ,  $Y = \{y_1, y_2, \dots, y_5\}$ , the set  $\mathcal{G}_{Aut} \subset \mathrm{Aut}(G)$ ; a linear congruence generator  $h$  of maximal periodic length.



---

## Encryption and Decryption Procedure:

1. Alice and Bob agree privately on the private parameters: A finitely generated free group  $F = \langle X \mid \quad \rangle$ ,  $X = \{x_1, x_2, \dots, x_q\}$ ,  $q \geq 2$ , a free subgroup  $F_U = \langle U \mid \quad \rangle$  with Nielsen reduced set  $U = \{u_1, u_2, \dots, u_{2N}\} \subset F$ , a faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ , now it is  $F_{U'} = \langle U' \mid \quad \rangle$ ,  $U' = \{V_1, V_2, \dots, V_{2N}\}$  with  $V_i = \varphi(u_i)$ , an assignment  $a_i \hat{=} V_j \iff j \equiv i \pmod{N}$  and a starting seed  $g_{\bar{\alpha}} \in \mathcal{G}_{Aut}$ .
2. Alice wants to transmit the message

$$S = s_1 s_2 \cdots s_z, \quad z \geq 1,$$

with  $s_i \in A$  to Bob.

- 2.1. Alice cuts the message into parts of  $\text{rank}(G) - 1 = 4$  letters

$$S = \underbrace{s_1 s_2 s_3 s_4}_{S_1} \mid \underbrace{s_5 s_6 s_7 s_8}_{S_2} \mid \cdots \mid \underbrace{s_{z-3} s_{z-2} s_{z-1} s_z}_{S_\beta},$$

with  $s_i \in \{a_1, a_2, \dots, a_N\}$ . To describe the procedure we assume that the letters in the parts  $S_i$  are pairwise different and  $z = 4 \cdot t$ ,  $t \in \mathbb{N}$ . Thus, it is  $\beta = t$ .

- 2.3. She writes  $S$  as a sequence  $S'$  of matrices with

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N},$$

it is

$$S' = \underbrace{V'_1 V'_2 V'_3 V'_4}_{S'_1} \mid \underbrace{V'_5 V'_6 V'_7 V'_8}_{S'_2} \mid \cdots \mid \underbrace{V'_{z-3} V'_{z-2} V'_{z-1} V'_z}_{S'_\beta},$$

with  $V'_i \in \{V_1, V_2, \dots, V_{2N}\}$ .

- 2.2. Alice needs for the encryption procedure  $\beta = \frac{z}{4}$  automorphisms  $g_i \in \mathcal{G}_{Aut}$ . With the help of the congruence generator  $h$  and the knowledge of  $g_{\bar{\alpha}}$  she gets the automorphisms  $g_{y_1}, g_{y_2}, \dots, g_{y_\beta}$ . It is  $y_1 = \bar{\alpha}, y_2 = h(y_1), \dots, y_\beta = h(y_{\beta-1})$ .
- 2.3. For each part  $S'_i$ ,  $1 \leq i \leq \beta$ , of the message Alice needs an additional matrix  $P_i$  with the property, that  $P_i \notin F_{U'} \subset \text{SL}(2, \mathbb{Q})$ , remember  $U' = \{\varphi(u_1), \varphi(u_2), \dots, \varphi(u_{2N})\}$ . We call the additional matrices  $P_i$  arbitrary ephemeral keys for Alice.
- 2.4. Alice encrypts the message as follows: for the part  $S'_1$  she applies the Nielsen transformation  $g_{x_1}$  to the set  $\{V'_1, V'_2, V'_3, V'_4, P_1\}$ , that is:

$$(V'_1, V'_2, V'_3, V'_4, P_1) \xrightarrow{g_{x_1}} (W_1, W_2, W_3, W_4, W_5),$$

with  $W_j$ ,  $1 \leq j \leq 5$ , words in  $\{V'_1, V'_2, V'_3, V'_4, P_1\}$ .

In general, she uses for the part  $S'_i$ ,  $1 \leq i \leq \beta$ , the automorphism  $g_{y_i}$  and the ephemeral key  $P_i$ , it is:

$$(V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i) \xrightarrow{g_{x_i}} (W_{5i-4}, W_{5i-3}, W_{5i-2}, W_{5i-1}, W_{5i}),$$

with  $W_j$ ,  $5i - 4 \leq j \leq 5i$ , words in  $\{V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i\}$ . Alice generates the ciphertext

$$C = W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} \cdots W_{z+\beta}$$

and sends it to Bob.

3. Bob gets the ciphertext

$$C = W_1W_2W_3W_4W_5W_6W_7W_8W_9W_{10} \cdots W_{z+\beta}$$

for decryption.

3.1. He cuts  $C$  into parts of 5 matrices

$$C = \underbrace{W_1W_2W_3W_4W_5}_{C'_1} \mid \underbrace{W_6W_7W_8W_9W_{10}}_{C'_2} \mid \cdots \mid \underbrace{W_{z+\beta-4} \cdots W_{z+\beta}}_{C'_\beta}.$$

3.2. With the knowledge of  $g_{\bar{\alpha}}$ , the linear congruence generator  $h$  and  $\beta (= \frac{z+\beta}{5})$  he computes for each automorphism  $g_{y_i} \in \mathcal{G}_{Aut}$ ,  $i = 1, 2, \dots, \beta$ , the inverse automorphism  $g_{y_i}^{-1}$ .

3.3 He applies on each ciphertext part  $C'_i$  the corresponding automorphism  $g_{y_i}^{-1}$ .  
For example

$$(W_1, W_2, W_3, W_4, W_5) \xrightarrow{g_{y_1}^{-1}} (V'_1, V'_2, V'_3, V'_4, P_1).$$

In general, for  $C'_i$ , it is

$$(W_{5i-4}, W_{5i-3}, W_{5i-2}, W_{5i-1}, W_{5i}) \xrightarrow{g_{y_i}^{-1}} (V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i).$$

Bob knows the set  $U'$  and hence he can decide which matrix in the reconstructed part is an element of the set  $U'$  and thus also that  $P_i \notin U'$ ,  $1 \leq i \leq \beta$ . Therefore, he knows that the last matrix in the reconstructed set is an ephemeral key of Alice and hence belongs not to the plaintext.

He gets the following sequence of matrices

$$S' = V'_1V'_2V'_3V'_4V'_5V'_6V'_7V'_8 \cdots V'_{z-3}V'_{z-2}V'_{z-1}V'_z,$$

with  $V'_i \in \{V_1, V_2, \dots, V_{2N}\}$ , and with the knowledge

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N}$$

he is able to read the plaintext

$$S = s_1s_2s_3s_4s_5s_6s_7s_8 \cdots s_{z-3}s_{z-2}s_{z-1}s_z,$$

with  $s_i \in A$ ,  $1 \leq i \leq z$ , from Alice.

The cryptographic protocol is summarized in Table 9.1 (page 219) and Table 9.2 (page 220).

Table 9.1.: Summary of **Protocol 10**: Private key cryptosystem using automorphisms on plaintext sequences I

<b>Public Knowledge</b>	
Abstract free group $G = \langle Y \mid \ \rangle$ , $Y = \{y_1, y_2, \dots, y_5\}$ ; plaintext alphabet $A = \{a_1, a_2, \dots, a_N\}$ with $N \geq 2$ ; subset $\mathcal{G}_{Aut} \subset Aut(G)$ ; linear congruence generator $h$ of maximal periodic length.	
<b>Alice</b>	<b>Bob</b>
Private keys	
Free group $F = \langle X \mid \ \rangle$ , $X = \{x_1, x_2, \dots, x_q\}$ , $q \geq 2$ , free subgroup $F_U = \langle U \mid \ \rangle$ of $F$ with Nielsen reduced set $U = \{u_1, u_2, \dots, u_{2N}\} \subset F$ ; faithful representation $\varphi : F \rightarrow SL(2, \mathbb{Q})$ ; $F_{U'} = \langle U' \mid \ \rangle$ , $U' = \{V_1, V_2, \dots, V_{2N}\}$ with $V_i = \varphi(u_i)$ ; assignment $a_i \hat{=} V_j \iff j \equiv i \pmod{N}$ and starting seed $g_{\bar{\alpha}} \in \mathcal{G}_{Aut}$ .	
Encryption	
Choose message $S = s_1 s_2 \cdots s_z$ , $z \geq 1$ and $s_i \in A$ .  Cut message into parts of $rank(G) - 1 = 4$ letters $S = \underbrace{s_1 s_2 s_3 s_4}_{S_1} \mid \underbrace{s_5 s_6 s_7 s_8}_{S_2} \mid \cdots \mid \underbrace{s_{z-3} s_{z-2} s_{z-1} s_z}_{S_\beta}$ .  Write $S$ as a sequence $S'$ of matrices with $a_i \hat{=} V_j \iff j \equiv i \pmod{N}$ , it is $S' = \underbrace{V'_1 V'_2 V'_3 V'_4}_{S'_1} \mid \underbrace{V'_5 V'_6 V'_7 V'_8}_{S'_2} \mid \cdots \mid \underbrace{V'_{z-3} V'_{z-2} V'_{z-1} V'_z}_{S'_\beta}$ , with $V'_i \in \{V_1, V_2, \dots, V_{2N}\}$ .  Calculate $\beta = \frac{z}{4}$ automorphisms $g_i \in \mathcal{G}_{Aut}$ . Compute $y_1 = \bar{\alpha}, y_2 = h(y_1), \dots, y_\beta = h(y_{\beta-1})$ and obtain $g_{y_1}, g_{y_2}, \dots, g_{y_\beta}$ .  For each part $S'_i$ , $1 \leq i \leq \beta$ , choose an additional matrix $P_i \in SL(2, \mathbb{Q})$ , with $P_i \notin F_{U'}$ , which is an ephemeral key.  Encryption: For $S'_i$ , $1 \leq i \leq \beta$ , choose ephemeral key $P_i$ and apply automorphism $g_{y_i}$ : $(V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i)$ $\downarrow g_{y_i}$ $(W_{5i-4}, W_{5i-3}, W_{5i-2}, W_{5i-1}, W_{5i})$ .  Generate ciphertext  $C = W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} \cdots W_{z+\beta}$  and send it to Bob.	$C = W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} \cdots W_{z+\beta}$ $\longrightarrow$

Table 9.2.: Summary of **Protocol 10**: Private key cryptosystem using automorphisms on plaintext sequences II

Alice	Bob
	Decryption
	<p>Cut <math>C</math> into parts of 5 matrices:  <math display="block">C = \underbrace{W_1W_2W_3W_4W_5}_{C'_1} \mid \underbrace{W_6W_7W_8W_9W_{10}}_{C'_2} \mid \cdots \mid \underbrace{W_{z+\beta-4} \cdots W_{z+\beta}}_{C'_\beta}.</math></p> <p>Compute <math>\beta</math> automorphisms:  <math>y_1 = \bar{\alpha}, y_2 = h(y_1), \dots, y_\beta = h(y_{\beta-1}),</math>  obtain <math>g_{y_1}, g_{y_2}, \dots, g_{y_\beta}.</math></p> <p>Compute for each automorphism <math>g_{y_i} \in \mathcal{G}_{Aut}, i = 1, 2, \dots, \beta,</math> the inverse automorphism <math>g_{y_i}^{-1}.</math></p> <p>Apply on each ciphertext part <math>C'_i</math> the corresponding automorphism <math>g_{y_i}^{-1}.</math> In general, for <math>C'_i,</math> it is:  <math display="block">\begin{array}{c} (W_{5i-4}, W_{5i-3}, W_{5i-2}, W_{5i-1}, W_{5i}) \\ \downarrow g_{y_i}^{-1} \\ (V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i) \end{array}</math></p> <p>Decide which matrices in the reconstructed part belong to the set <math>U'</math> and which not.</p> <p>Therefore, get sequence of matrices</p> $S' = V'_1V'_2V'_3V'_4V'_5V'_6V'_7V'_8 \cdots V'_{z-3}V'_{z-2}V'_{z-1}V'_z,$ <p>with <math>V'_i \in \{V_1, V_2, \dots, V_{2N}\},</math> and with the knowledge</p> $a_i \hat{=} V_j \iff j \equiv i \pmod{N}$ <p>read the plaintext</p> $S = s_1s_2s_3s_4s_5s_6s_7s_8 \cdots s_{z-3}s_{z-2}s_{z-1}s_z$ <p>from Alice.</p>

**Remark 9.0.1.** The ephemeral key  $P_i$  for the sequence  $(V'_{4i-3}, V'_{4i-2}, V'_{4i-1}, V'_{4i}, P_i)$  must not be necessarily at the end, Alice can write it at each position, which she wants. For example  $(V'_{4i-3}, P_i, V'_{4i-2}, V'_{4i-1}, V'_{4i}).$  Bob reconstructs exactly the sequence, because he knows the set  $U'.$  He is able to decide which element in  $(V'_{4i-3}, P_i, V'_{4i-2}, V'_{4i-1}, V'_{4i})$  is not an element of  $U'$  and therefore he knows, that  $P_i$  is an ephemeral key from Alice and belongs not to the plaintext, thus he can delete this matrix.

To describe this scheme we assumed that  $z = 4 \cdot t,$  with  $t \in \mathbb{N},$  and that the letters in each part  $S_i$  are pairwise different. We now look at the situations, when  $z \in \mathbb{N}$  is not necessarily  $z = 4 \cdot t,$   $t \in \mathbb{N},$  and the letters in each part  $S_i$  are not necessarily pairwise different.

**Remark 9.0.2.** Let

$$S = V'_1 V'_2 \cdots V'_z, \quad \text{with } z \geq 1, V'_i \in U',$$

be a plaintext written as a sequence of matrices.

There are two cases, which differ to the assumptions for the cryptographic protocol above:

1. The first one is, that  $z \in \mathbb{N}$  is not necessarily  $z = 4 \cdot t$ ,  $t \in \mathbb{N}$ . Thus, Alice needs  $\beta = \lceil \frac{z}{4} \rceil$  (for  $r \in \mathbb{R}$  it is  $\lceil r \rceil := \min\{x \in \mathbb{Z} | x \geq r\}$ ) automorphisms of  $\mathcal{G}_{Aut}$  for encryption. Moreover, Alice adds  $r' = 4 - r$ , with  $r \equiv z \pmod{4}$  and  $0 \leq r' \leq 3$ , matrices  $P_j \in \text{SL}(2, \mathbb{Q})$ , with  $P_j \notin F_{U'}$  to  $S$ . Bob knows  $U'$  and hence he is able to identify the added matrices from Alice and knows these matrices belong not to the plaintext. Alice can add the matrices  $P_j$  on an arbitrary place in the plaintext.

2. The second one is, that the letters in a part  $S_i$  of a message

$$S = \underbrace{s_1 s_2 s_3 s_4}_{S_1} | \underbrace{s_5 s_6 s_7 s_8}_{S_2} | \cdots | \underbrace{s_{z-3} s_{z-2} s_{z-1} s_z}_{S_\beta}$$

are not necessarily pairwise different. Therefore, we get two cases.

- 2.1. 1. case: One letter appears twice in the plaintext sequence  $S_i$ :

$$S_i = \{s_{4i-3}, s_{4i-2}, s_{4i-1}, s_{4i}\}.$$

For example it is  $s_{4i-2} = s_{4i-1} = a_t$ ,  $s_{4i-3} = a_k$  and  $s_{4i} = a_r$ , with  $a_t, a_k, a_r \in \{a_1, a_2, \dots, a_N\}$  and pairwise different.

As above Alice writes the plaintext as a sequence of matrices in  $U'$ , it is  $F_{U'} = \langle U' | \ \rangle$ , with  $U' = \{V_1, V_2, \dots, V_{2N}\}$  and

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N}.$$

Alice chooses  $a_t = s_{4i-2} \hat{=} V_t$  and  $a_r = s_{4i-1} \hat{=} V_{N+t}$ . Therefore, it is  $V_t \neq V_{N+t}$  and the cryptosystem is feasible. Thus, she gets

$$S'_i = \{V_k, V_t, V_{N+t}, V_r\}, \quad V_i \in U'.$$

- 2.2. 2. case: One letter appears three or four times in the plaintext sequence  $S_i$ :

$$S_i = \{s_{4i-3}, s_{4i-2}, s_{4i-1}, s_{4i}\}.$$

Split  $S_i$  into two parts

$$S_{i_1} = \{s_{4i-3}, s_{4i-2}, P_j, P_{j+1}\} \text{ and } S_{i_2} = \{s_{4i-1}, s_{4i}, P_{j+2}, P_{j+3}\}.$$

Let  $P_j, \dots, P_{j+3} \in \text{SL}(2, \mathbb{Q})$  be pairwise different matrices, a basis for a subgroup of rank 4, and  $P_j, \dots, P_{j+3} \notin F_{U'}$ . Alice writes the plaintext as a sequence of matrices. If she gets for  $S_{i_1}$  or  $S_{i_2}$  the situation as in point 2.1, she acts as it is described there. Now, it is

$$S'_i = \{V'_\ell, V'_{\ell+1}, V'_{\ell+2} = P_j, V'_{\ell+3} = P_{j+1}\} \text{ and } S'_{i+1} = \{V'_{\ell+4}, V'_{\ell+5}, V'_{\ell+6} = P_j, V'_{\ell+7} = P_{j+1}\}$$

and the matrices in each set are pairwise different.

Bob is able to identify the matrices  $P_i$  which do not belong to the plaintext and decrypt the message correctly because he knows the set  $U'$ .

Alice has to calculate ephemeral matrices in  $SL(2, \mathbb{Q})$ , we now give proposals how Alice could generate these matrices, which are not matrices in  $F_U$ .

**Remark 9.0.3.** We give three opportunities to generate ephemeral matrices for Alice, the third one is a mix of the first and the second one.

1. Alice knows the Nielsen reduced set  $U = \{u_1, u_2, \dots, u_{2N}\}$ . She generates a Nielsen reduced set  $U_A = U \cup \{u_{2N+1}, u_{2N+2}, \dots, u_{2N+t}\}$ , with  $u_j$ ,  $2N+1 \leq j \leq 2N+t$ , words in  $X$  and  $t$  the minimum of ephemeral matrices which she needs for encryption of one sequence  $S_j$  of the plaintext. Her ephemeral matrices are  $\varphi(u_{2N+1}), \varphi(u_{2N+2}), \dots, \varphi(u_{2N+t})$ .
2. Alice chooses an abstract set  $X_A = \{x_{q+1}, x_{q+2}, \dots, x_{q+p}\}$ , with  $p \geq 2$  and  $X \cap X_A = \emptyset$ , and generates a Nielsen reduced set  $U_A = \{u_{2N+1}, u_{2N+2}, \dots, u_{2N+t}\}$ , with  $u_j$ ,  $2N+1 \leq j \leq 2N+t$ , words in  $X_A$  and  $t$  the minimum of ephemeral matrices which she needs for encryption of one sequence  $S_j$  of the plaintext. She chooses a faithful representation  $\varphi_A$  into  $SL(2, \mathbb{Q})$ , with the property  $\varphi_A(x_i) = \varphi(x_i) = M_i$  for all  $i = 1, 2, \dots, q$ , and  $\varphi_A(x_{q+s}) = N_s$  with  $1 \leq s \leq p$ . The set  $\{M_1, M_2, \dots, M_q\} \cup \{N_1, N_2, \dots, N_p\}$  is a free generating set for a free subgroup in  $SL(2, \mathbb{Q})$  of rank  $q+p$ . To generate these matrices she can use Theorem 4.2.18. Her ephemeral matrices are  $\varphi_A(u_{2N+1}), \varphi_A(u_{2N+2}), \dots, \varphi_A(u_{2N+t})$ .
3. As in 2. Alice chooses an abstract set  $X_A = \{x_{q+1}, x_{q+2}, \dots, x_{q+p}\}$ , with  $p \geq 2$  and  $X \cap X_A = \emptyset$ , she then generates a Nielsen reduced set  $U_A = U \cup \{u_{2N+1}, u_{2N+2}, \dots, u_{2N+t}\}$ , with  $u_j$ ,  $2N+1 \leq j \leq 2N+t$ , words in  $X \cup X_A$  and  $t$  the minimum of ephemeral matrices which she needs for encryption of one sequence  $S_j$  of the plaintext. She chooses a faithful representation  $\varphi_A$  into  $SL(2, \mathbb{Q})$ , with the property  $\varphi_A(x_i) = \varphi(x_i) = M_i$  for all  $i = 1, 2, \dots, q$ , and  $\varphi_A(x_{q+s}) = N_s$  with  $1 \leq s \leq p$ . The set  $\{M_1, M_2, \dots, M_q\} \cup \{N_1, N_2, \dots, N_p\}$  is a free generating set for a free subgroup in  $SL(2, \mathbb{Q})$  of rank  $q+p$ . To generate these matrices she can use Theorem 4.2.18. Her ephemeral matrices are  $\varphi_A(u_{2N+1}), \varphi_A(u_{2N+2}), \dots, \varphi_A(u_{2N+t})$ .

We explained **Protocol 10** by fixing the rank of the abstract group  $G$ ,  $rank(G) = 5$ , and the cardinality of  $U$ ,  $|U| = 2N$ . This helped us to explain this private key cryptosystem, but it is not mandatory. We can variate these values.

**Variation 9.0.4.** 1. We can choose a finitely generated abstract free group  $G$  with a rank greater than or equal to 2; another option is to choose different finitely generated free groups  $G_i$  with pairwise different ranks  $rank(G_i) \geq 2$ . The set  $\mathcal{G}_{Aut}$  from which Alice and Bob get the automorphisms for encryption and decryption, respectively, is then a subset of  $\bigcup_i Aut(G_i)$ .

2. The set  $U$  can be chosen with cardinality  $|U| = k \cdot N$ ,  $k \geq 2$ . Each sequence  $S_i$  which we get by cutting the plaintext into pieces must, in general, have a length between 1 and  $rank(G_j) - 1$ , depending on the group  $G_j$  on which the automorphism acts for the sequence  $S_i$ , because we now add to each sequence at least one ephemeral matrix. We have to take care that the elements in each sequence  $S_i$  with the additional ephemeral keys form a basis for a free group of  $rank(G_j)$ . For this it is possible to align the set  $U$ , that means choose  $k$  with  $k \geq \max\{rank(G_i)\} - 1$  for  $|U| = k \cdot N$  and act like in Remark 9.0.2 case 2.1. Another option<sup>i</sup> is to split the sequence with more than  $k$  identical letters in a similar way as explained in Remark 9.0.2 case 2.2. The ephemeral keys in each sequence must be pairwise different and after construction, see Remark 9.0.3, they are elements of a basis.

We now take a look at the security.

---

**Security 9.0.5.** Let

$$C = C_1 C_2 C_3 \cdots C_{z+\beta},$$

with  $C_i \in \text{SL}(2, \mathbb{Q})$ , be a ciphertext to a plaintext with  $z$  letters, and  $\beta$  is the number of added ephemeral matrices by Alice. The ciphertext is longer than the plaintext. If Alice takes care, that each plaintext sequence  $C_i$  has as one product at least one ephemeral matrix  $P_j$  (that means it is not only written as a word in  $U'$ ), then  $C_i \notin F_{U'}$  and hence each plaintext sequence is encrypted differently, even if the same plaintext with the same automorphisms is used and only the ephemeral matrices from Alice are changed. There exist infinitely many ephemeral matrices for Alice. Therefore, the cryptosystem is a polyalphabetic system, that means, a matrix  $V_i \in U'$ , and hence a letter  $a_i \in A$ , is encrypted differently at different positions in the plaintext. Hence, a statistical frequency attack (see for instance [BFKR15]), for the ciphertext, over the frequency of matrices, which corresponds to letters in the plaintext alphabet, or groups of words, is useless. It is very unlikely that Eve makes a correct guess for the set  $U'$ , of the used matrices to write the ciphertext matrices  $C_j$ ,  $1 \leq j \leq b+z$ , because Eve does not know the number  $q$  which gives the number of matrices of the form

$$M_j = \begin{pmatrix} -r_j & -1 + r_j^2 \\ 1 & -r_j \end{pmatrix} \quad (9.1)$$

which are the basic to generate the matrices in  $U'$ .

Alice and Bob are working with  $\varphi(u_i)$ ,  $u_i$  words in  $X$ , instead with  $\varphi(x_j) = M_j$ , because if they generate the matrices  $\varphi(x_j) = M_j$  with Theorem 4.2.18 the matrices are of a special form (9.1). Then, maybe, there are attacks possible for this cryptosystem, because an eavesdropper knows when she gets likely the correct automorphism of the set  $\mathcal{G}_{Aut}$  by the form (9.1) of the matrices. Even if Eve makes a correct guess for the set  $U'$  she cannot use this set to solve the constructive membership problem for the matrices  $C_j$ , because for this problem there is no algorithm known for the used matrix group, and if Alice takes care that in each  $C_j$  is at least one ephemeral matrix contained, then Eve needs also this ephemeral matrix to write  $C_j$  as a word in  $U'$  in conjunction with this ephemeral matrix.

Eve could do Nielsen transformations, which are the inverse automorphisms of  $\mathcal{G}_{Aut}$ , on the sequences of ciphertext matrices and tries to get matrices of  $U' = \{\varphi(u_1), \varphi(u_2), \dots, \varphi(u_{2N})\}$ . Especially, if we assume that Eve knows how to cut the ciphertext into the correct sequences, in the case where the abstract groups  $G_i$  are of different rank (see Variation 9.0.4). This is a **brute force search** through the inverse elements of the elements of  $\mathcal{G}_{Aut}$ . In general, she does not know how  $U'$  looks like. Thus, even if she generates matrices she cannot be sure if these are matrices of  $U'$  she is also not able to identify the ephemeral added matrices of Alice. She knows that in each sequence is at least one ephemeral matrix. Assume Eve is able to generate the correct sequence of plaintext matrices with inverse automorphisms of the set  $\mathcal{G}_{Aut}$ , she then has to do a statistical frequency attack (see for instance [BFKR15]) to decrypt the plaintext, because she does not now which matrix belongs to which letter in the plaintext alphabet. The added ephemeral matrices could confuse this statistical frequency attack, if Alice uses her ephemeral matrices more than once but in different plaintext sequences.

Due to the fact, that there is no algorithm known to solve the constructive membership problem for free (discrete) subgroups of  $\text{SL}(2, \mathbb{Q})$  of rank greater than or equal to 2, Eve cannot use the set  $U'$ , if she gets it, to rewrite for the next ciphertext the element  $C_j$  as a word in  $U'$  and hence she cannot get hints for the used automorphisms, in particular she does not know the ephemeral matrices from Alice.

We present a second variation, in which the used automorphisms are private.

**Variation 9.0.6.** Instead of the public set  $\mathcal{G}_{Aut}$  and the linear congruence generator, Alice and Bob agree privately on two or more abstract free groups  $G_i$  with  $rank(G_i) \geq 2$ ,  $i \geq 2$ . They choose privately two or more automorphisms  $g_{\ell_k} \in \bigcup_{i=1}^j Aut(G_i)$ , with  $g_{\ell_k} \in Aut(G_k)$ . For example it is  $j = 2$  and they choose  $t$  automorphisms for the encryption and decryption

$$\begin{aligned} g_{1_1} &\in Aut(G_1), \\ g_{2_1} &\in Aut(G_1), \\ g_{3_2} &\in Aut(G_2), \\ &\vdots \\ g_{t_1} &\in Aut(G_1). \end{aligned}$$

Alice writes as before the plaintext  $S$  as a sequence of matrices

$$S' = V'_1 V'_2 V'_3 V'_4 V'_5 V'_6 V'_7 V'_8 \cdots V'_{z-3} V'_{z-2} V'_{z-1} V'_z.$$

Then, she cuts  $S'$  into pieces with  $|S'_i| < rank(G_j)$ , because  $g_{i_j} \in Aut(G_j)$  and the encryption of the part  $S'_i$  is  $g_{i_j}(S'_i \cup N'_i)$  with  $N'_i$  a set of ephemeral matrices for Alice with  $|N'_i| = rank(G_j) - |S'_i|$  and for each  $P_\ell \in N'_i$  it is  $P_\ell \notin F_{U'}$ .

If the number of automorphisms ends before all of the plaintext  $S$  is encrypted, Alice starts again from the beginning of the automorphisms (which is  $g_{1_1} \in Aut(G_1)$ ) and uses the automorphisms in the same order again.

Bob is able to encrypt the ciphertext because he knows the set  $U'$  and also the automorphisms  $g_{i_j} \in Aut(G_j)$  and hence the corresponding inverse automorphisms  $g_{i_j}^{-1} \in Aut(G_j)$ .

Because of the fact, that the automorphisms are private an eavesdropper, Eve, does not know how long the sequences are in which the plaintext is cut. Thus, Eve does not know which matrices form a set on which she should do Nielsen transformations. She does also not know which inverse automorphisms she could use for a brute force search. Hence, she has to do Nielsen transformations on a sequence and tries to get matrices, but she does not know when she gets the correct matrices and hence when to stop with her Nielsen transformations.

We calculate an example for this last variation.

**Example 9.0.7.** We perform the steps for an example of a symmetric key cryptosystem with Variation 9.0.6 and the help of the computer program Maple 16 and GAP, see Appendix C.10 for details.

We first give the public information. Let  $F$  be a free group of rank 4 with free generating set  $X = \{a, b, c, d\}$  and let  $A := \{B, I, E, L, K, O, M\}$  be the plaintext alphabet, hence it is  $N := |A| = 7$ .

For the private parameters Alice and Bob agree on the following common secret keys:

1. Because of Variation 9.0.6, they choose two abstract free groups. The first one is  $G_1$  with  $rank(G_1) = 5$ . Therefore, it is

$$G_1 = \langle x_1, x_2, x_3, x_4, x_5 \mid \rangle$$

and the second one is  $G_2$  with  $rank(G_2) = 4$ , hence

$$G_2 = \langle y_1, y_2, y_3, y_4 \mid \rangle.$$



They choose for each group one automorphism, it is  $g_{1_1} \in \text{Aut}(G_1)$  and  $g_{2_2} \in \text{Aut}(G_2)$ . The first one can be described with Nielsen transformations as follows

$$(N1)_1 (N1)_4 (N2)_{2.3} (N2)_{5.3} (N2)_{1.3} (N2)_{4.2} (N1)_5 (N2)_{1.2} (N2)_{2.4} (N2)_{3.1},$$

hence we get the automorphism

$$\begin{aligned} g_{1_1} : G_1 &\rightarrow G_1 \\ x_1 &\mapsto x_1^{-1} x_3 x_2 x_3, \\ x_2 &\mapsto x_2 x_3 x_4^{-1} x_2 x_3, \\ x_3 &\mapsto x_3 x_1^{-1} x_3 x_2 x_3, \\ x_4 &\mapsto x_4^{-1} x_2 x_3, \\ x_5 &\mapsto x_3^{-1} x_5^{-1} x_3. \end{aligned}$$

The second automorphism is describable with Nielsen transformations as follows

$$[(N2)_{3.1}]^2 (N1)_2 [(N2)_{2.1}]^3 (N2)_{2.4} (N2)_{4.2} (N2)_{1.3}.$$

Therefore, the automorphism is

$$\begin{aligned} g_{2_2} : G_2 &\rightarrow G_2 \\ y_1 &\mapsto y_1 y_3 y_1^2, \\ y_2 &\mapsto y_2^{-1} y_1^3 y_4, \\ y_3 &\mapsto y_3 y_1^2, \\ y_4 &\mapsto y_4 y_2^{-1} y_1^3 y_4. \end{aligned}$$

2. The alphabet  $A$  consists of  $N = 7$  elements, hence Alice and Bob choose a Nielsen reduced set with  $2N = 14$  elements. Thus, let  $U = \{u_1, u_2, \dots, u_{14}\}$  be a subset of  $F$ , which is Nielsen reduced and

$$\begin{aligned} u_1 &:= ba^2, & u_8 &:= bc^{-1}bab^{-1}, \\ u_2 &:= cd, & u_9 &:= c^2ba, \\ u_3 &:= d^2c^{-2}, & u_{10} &:= c^2dab^{-1}, \\ u_4 &:= a^{-1}b, & u_{11} &:= dabd^{-1}a, \\ u_5 &:= a^4b^{-1}, & u_{12} &:= a^{-1}d^3c^{-1}, \\ u_6 &:= b^3a^{-2}, & u_{13} &:= a^{-1}c^{-1}bac^{-2}, \\ u_7 &:= bc^3, & u_{14} &:= a^2db^2d^{-1}. \end{aligned}$$

3. They agree on the faithful representation

$$\begin{aligned}\varphi : F &\rightarrow H \subset \text{SL}(2, \mathbb{Q}) \\ a &\mapsto \begin{pmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{pmatrix}, \\ b &\mapsto \begin{pmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{pmatrix}, \\ c &\mapsto \begin{pmatrix} \frac{-23}{2} & \frac{525}{4} \\ 1 & \frac{-23}{2} \end{pmatrix}, \\ d &\mapsto \begin{pmatrix} \frac{-35}{2} & \frac{1221}{4} \\ 1 & \frac{-35}{2} \end{pmatrix}.\end{aligned}$$

The set  $M = \left\{ \begin{pmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{pmatrix}, \begin{pmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{pmatrix}, \begin{pmatrix} \frac{-23}{2} & \frac{525}{4} \\ 1 & \frac{-23}{2} \end{pmatrix}, \begin{pmatrix} \frac{-35}{2} & \frac{1221}{4} \\ 1 & \frac{-35}{2} \end{pmatrix} \right\}$  is a free generating set for a free subgroup  $H$  of  $\text{SL}(2, \mathbb{Q})$ , because of Theorem 4.2.18.

They generate the set  $U' = \{V_1, V_2, \dots, V_{14}\}$ , with  $V_i = \varphi(u_i)$ , and because of the assignment

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N}$$

they know

$$\begin{aligned}V_1 &:= \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix}, & V_8 &:= \begin{pmatrix} \frac{10733}{2} & \frac{155745}{4} \\ -691 & \frac{-10027}{2} \end{pmatrix} && \hat{=} \mathbf{B}; \\ V_2 &:= \begin{pmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{pmatrix}, & V_9 &:= \begin{pmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{pmatrix} && \hat{=} \mathbf{I}; \\ V_3 &:= \begin{pmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{pmatrix}, & V_{10} &:= \begin{pmatrix} -647496 & \frac{-9392507}{2} \\ 56518 & 409922 \end{pmatrix} && \hat{=} \mathbf{E}; \\ V_4 &:= \begin{pmatrix} 15 & -109 \\ 4 & -29 \end{pmatrix}, & V_{11} &:= \begin{pmatrix} 563077 & -2011276 \\ -32264 & 115245 \end{pmatrix} && \hat{=} \mathbf{L}; \\ V_5 &:= \begin{pmatrix} -4575 & -33209 \\ 1364 & 9901 \end{pmatrix}, & V_{12} &:= \begin{pmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{pmatrix} && \hat{=} \mathbf{K}; \\ V_6 &:= \begin{pmatrix} \frac{95009}{2} & \frac{638869}{4} \\ -6391 & \frac{-42975}{2} \end{pmatrix}, & V_{13} &:= \begin{pmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{pmatrix} && \hat{=} \mathbf{O}; \\ V_7 &:= \begin{pmatrix} \frac{149079}{2} & \frac{-3415829}{4} \\ -10009 & \frac{229335}{2} \end{pmatrix}, & V_{14} &:= \begin{pmatrix} \frac{3682603}{2} & \frac{128159475}{4} \\ -548633 & \frac{-19093157}{2} \end{pmatrix} && \hat{=} \mathbf{M}.\end{aligned}$$

We look at the encryption and decryption procedure for Alice and Bob.

1. With the above agreements Alice is able to encrypt her message

$$S = \text{ILIKEBOB}.$$

a) Firstly, she writes the message  $S$  as sequence of matrices

$$\begin{aligned}
S' &= V_2 V_4 V_9 V_{12} V_3 V_1 V_{13} V_1 \\
&= \begin{pmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{pmatrix} \begin{pmatrix} 15 & -109 \\ 4 & -29 \end{pmatrix} \begin{pmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{pmatrix} \\
&\quad \begin{pmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{pmatrix} \begin{pmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{pmatrix} \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix} \\
&\quad \begin{pmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{pmatrix} \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix}.
\end{aligned}$$

b) Secondly, she cuts the matrix-plaintext  $S'$  into pieces to apply for encryption the automorphisms  $g_{1_1} \in \text{Aut}(G_1)$ , with  $\text{rank}(G_1) = 5$  and  $g_{2_2} \in \text{Aut}(G_2)$ , with  $\text{rank}(G_2) = 4$ . Alice gets:

$$\begin{aligned}
S'_1 &= \left\{ \begin{pmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{pmatrix}, \begin{pmatrix} 15 & -109 \\ 4 & -29 \end{pmatrix}, \begin{pmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{pmatrix}, \begin{pmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{pmatrix} \right\} \\
&\Rightarrow \left( \begin{pmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{pmatrix}, \begin{pmatrix} 15 & -109 \\ 4 & -29 \end{pmatrix}, \begin{pmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{pmatrix}, \begin{pmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{pmatrix} \right) \hat{=} \text{ILIK} \\
S'_2 &= \left\{ \begin{pmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{pmatrix}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix} \right\} \\
&\Rightarrow \left( \begin{pmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{pmatrix}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix} \right) \hat{=} \text{EB} \\
S'_3 &= \left\{ \begin{pmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{pmatrix}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix} \right\} \\
&\Rightarrow \left( \begin{pmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{pmatrix}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix} \right) \hat{=} \text{OB}
\end{aligned}$$

c) Thirdly, Alice needs additional matrices  $P_j \notin F_{U'}$ , as arbitrary ephemeral keys.

- For the first sequence  $S'_1$ , she needs one ephemeral matrix, because  $|S'_1| = 4$  and she applies  $g_{1_1}$  on this set and it is  $\text{rank}(G_1) = 5$ .
- For the second sequence  $S'_2$ , she needs two ephemeral matrices, because  $|S'_2| = 2$  and she applies  $g_{2_2}$  on this set and it is  $\text{rank}(G_2) = 4$ .
- For the third sequence  $S'_3$ , she needs three ephemeral matrices, because  $|S'_3| = 2$  and she applies  $g_{1_1}$  on this set and it is  $\text{rank}(G_1) = 5$ .

To generate ephemeral matrices, Alice uses the method explained in 2. of Remark 9.0.3. She needs at most three ephemeral keys in each sequence  $S'_i$ .

Thus, Alice chooses the set  $X_A = \{z_5, z_6, z_7\}$  and extends  $\varphi$  to  $\varphi_A$  with

$$\begin{aligned}\varphi_A : F &\rightarrow \text{SL}(2, \mathbb{Q}) \\ a &\mapsto \begin{pmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{pmatrix}, \\ b &\mapsto \begin{pmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{pmatrix}, \\ c &\mapsto \begin{pmatrix} \frac{-23}{2} & \frac{525}{4} \\ 1 & \frac{-23}{2} \end{pmatrix}, \\ d &\mapsto \begin{pmatrix} \frac{-35}{2} & \frac{1221}{4} \\ 1 & \frac{-35}{2} \end{pmatrix}, \\ z_5 &\mapsto N_1 := \begin{pmatrix} \frac{-43}{2} & \frac{1845}{4} \\ 1 & \frac{-43}{2} \end{pmatrix}, \\ z_6 &\mapsto N_2 := \begin{pmatrix} \frac{-55}{2} & \frac{3021}{4} \\ 1 & \frac{-55}{2} \end{pmatrix}, \\ z_7 &\mapsto N_3 := \begin{pmatrix} \frac{-63}{2} & \frac{3965}{4} \\ 1 & \frac{-63}{2} \end{pmatrix}.\end{aligned}$$

After Theorem 4.2.18  $\varphi_A(a)$ ,  $\varphi_A(b)$ ,  $\varphi_A(c)$ ,  $\varphi_A(d)$ ,  $\varphi_A(z_5)$ ,  $\varphi_A(z_6)$  and  $\varphi_A(z_7)$  together generate a free subgroup of  $\text{SL}(2, \mathbb{Q})$  with rank 7.

To get a subgroup of rank 4 she generates a Nielsen reduced set with 4 elements, it is

$$N' = \{N_1N_2^2, N_2N_3, N_3N_1^2, N_1^{-1}N_2N_1N_2\}.$$

She gets

$$\begin{aligned}N'_1 = N_1N_2^2 &= \begin{pmatrix} -57866 & \frac{3180525}{2} \\ 2694 & -74036 \end{pmatrix}, & N'_2 = N_2N_3 &= \begin{pmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{pmatrix}, \\ N'_3 = N_3N_1^2 &= \begin{pmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{pmatrix}, & N'_4 = N_1^{-1}N_2N_1N_2 &= \begin{pmatrix} \frac{621893}{2} & \frac{-34178721}{4} \\ 14351 & \frac{-788719}{2} \end{pmatrix}.\end{aligned}$$

Hence, her ephemeral keys are

$$P_1 = N'_1, \quad P_2 = N'_2, \quad P_3 = N'_3 \quad \text{and} \quad P_4 = N'_4.$$

Now, Alice is able to encrypt her message

- She adds to the first sequence  $S'_1 \hat{=} \text{ILIK}$  the ephemeral key  $P_1$  and applies the automorphism  $g_{1_1}$  on

$$\left( \begin{pmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{pmatrix}, \underbrace{\begin{pmatrix} -57866 & \frac{3180525}{2} \\ 2694 & -74036 \end{pmatrix}}_{=P_1}, \begin{pmatrix} 15 & -109 \\ 4 & -29 \end{pmatrix}, \begin{pmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{pmatrix}, \begin{pmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{pmatrix} \right).$$

She gets

$$\left( \left( \begin{pmatrix} \frac{453037463005}{2} & \frac{-6566656978411}{4} \\ 12969541169 & \frac{-187990033891}{2} \end{pmatrix}, \begin{pmatrix} \frac{-515958453260453803}{2} & \frac{7478679920196901999}{4} \\ 12010438543010031 & \frac{-174088097591505391}{2} \end{pmatrix}, \right. \\ \left. \begin{pmatrix} \frac{3968201970233}{2} & \frac{-57518027287927}{4} \\ 529958232109 & \frac{-7681602973983}{2} \end{pmatrix}, \begin{pmatrix} \frac{83406030953}{2} & \frac{-1208948133265}{4} \\ 12234456659 & \frac{-177335180327}{2} \end{pmatrix}, \begin{pmatrix} \frac{-65207575}{2} & \frac{991192539}{4} \\ -4494561 & \frac{68319905}{2} \end{pmatrix} \right).$$

- She adds to the second sequence  $S'_2 \hat{=} \text{EB}$  the ephemeral keys  $P_2$  and  $P_3$ , then she applies the automorphism  $g_{2_2}$  on

$$\left( \begin{pmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{pmatrix}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix}, \underbrace{\begin{pmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{pmatrix}}_{=P_2}, \underbrace{\begin{pmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{pmatrix}}_{=P_3} \right).$$

She gets

$$\left( \begin{pmatrix} \frac{-1104332496534507861}{2} & \frac{-25304312660337129571}{4} \\ 31604200843034185 & \frac{724168293536436571}{2} \end{pmatrix}, \begin{pmatrix} -480689945680474129277 & 10323650084255317045974 \\ -143263719821090419728 & 3076836797799093562123 \end{pmatrix}, \begin{pmatrix} \frac{22386390293811}{2} & \frac{512954405587601}{4} \\ -407276382779 & \frac{-9332197468925}{2} \end{pmatrix}, \begin{pmatrix} -186180075388817073675749582 & \frac{7997079898367833227219056023}{2} \\ 5914022532266907628279666 & -127013888603589241202576880 \end{pmatrix} \right).$$

- She adds to the third sequence  $S'_3 \hat{=} \text{OB}$  the ephemeral keys  $P_4$ ,  $P_2$  and  $P_3$ , then she applies the automorphism  $g_{3_1}$ , which is  $g_{1_1}$  because Alice and Bob only agree on two automorphisms, on

$$\left( \begin{pmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{pmatrix}, \underbrace{\begin{pmatrix} \frac{621893}{2} & \frac{-34178721}{4} \\ 14351 & \frac{-788719}{2} \end{pmatrix}}_{=P_4}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix}, \right. \\ \left. \underbrace{\begin{pmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{pmatrix}}_{=P_2}, \underbrace{\begin{pmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{pmatrix}}_{=P_3} \right).$$

She gets

$$\begin{pmatrix} \left( \begin{array}{cc} \frac{-1616087435846771117}{2} & \frac{10844781227098250059}{2} \\ 70532776776146599 & \frac{-473311354639843285}{2} \end{array} \right), \\ \left( \begin{array}{cc} \frac{5117735040480436319307}{2} & \frac{-34342644872543531950151}{2} \\ 118098476048874935309 & \frac{-792501759245893528165}{2} \end{array} \right), \\ \left( \begin{array}{cc} \frac{1176330057042013989893}{2} & \frac{-7893782128685642713947}{2} \\ -79397180640094685191 & \frac{532796082062893539917}{2} \end{array} \right), \\ \left( \begin{array}{cc} -3473922528580 & \frac{23311814068153}{2} \\ -110342647234 & 370228071430 \end{array} \right), \\ \left( \begin{array}{cc} 30697842540 & \frac{-205999121749}{2} \\ 9149177258 & -30697963178 \end{array} \right) \end{pmatrix}.$$

Alice sends the ciphertext

$$\begin{aligned} C = & C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12} C_{13} C_{14} \\ & \left( \begin{array}{cc} \frac{453037463005}{2} & \frac{-6566656978411}{2} \\ 12969541169 & \frac{-187990033891}{2} \end{array} \right), \left( \begin{array}{cc} \frac{-515958453260453803}{2} & \frac{7478679920196901999}{2} \\ 12010438543010031 & \frac{-174088097591505391}{2} \end{array} \right), \\ & \left( \begin{array}{cc} \frac{3968201970233}{2} & \frac{-57518027287927}{2} \\ 529958232109 & \frac{-7681602973983}{2} \end{array} \right), \left( \begin{array}{cc} \frac{83406030953}{2} & \frac{-1208948133265}{2} \\ 12234456659 & \frac{-177335180327}{2} \end{array} \right), \left( \begin{array}{cc} \frac{-65207575}{2} & \frac{991192539}{2} \\ -4494561 & \frac{68319905}{2} \end{array} \right) \\ & \left( \begin{array}{cc} \frac{-1104332496534507861}{2} & \frac{-25304312660337129571}{2} \\ 31604200843034185 & \frac{724168293536436571}{2} \end{array} \right), \\ & \left( \begin{array}{cc} -480689945680474129277 & 10323650084255317045974 \\ -143263719821090419728 & 3076836797799093562123 \end{array} \right), \\ & \left( \begin{array}{cc} \frac{22386390293811}{2} & \frac{512954405587601}{2} \\ -407276382779 & \frac{-9332197468925}{2} \end{array} \right), \\ & \left( \begin{array}{cc} -186180075388817073675749582 & \frac{7997079898367833227219056023}{2} \\ 5914022532266907628279666 & -127013888603589241202576880 \end{array} \right) \\ & \left( \begin{array}{cc} \frac{-1616087435846771117}{2} & \frac{10844781227098250059}{2} \\ 70532776776146599 & \frac{-473311354639843285}{2} \end{array} \right), \\ & \left( \begin{array}{cc} \frac{5117735040480436319307}{2} & \frac{-34342644872543531950151}{2} \\ 118098476048874935309 & \frac{-792501759245893528165}{2} \end{array} \right), \\ & \left( \begin{array}{cc} \frac{1176330057042013989893}{2} & \frac{-7893782128685642713947}{2} \\ -79397180640094685191 & \frac{532796082062893539917}{2} \end{array} \right), \\ & \left( \begin{array}{cc} -3473922528580 & \frac{23311814068153}{2} \\ -110342647234 & 370228071430 \end{array} \right), \\ & \left( \begin{array}{cc} 30697842540 & \frac{-205999121749}{2} \\ 9149177258 & -30697963178 \end{array} \right) \end{aligned}$$

to Bob.

Bob knows the automorphisms  $g_1 \in \text{Aut}(G_1)$  and  $g_2 \in \text{Aut}(G_2)$  and also the rank of  $G_1$ , which is  $\text{rank}(G_1) = 5$ , and the rank of  $G_2$ , which is  $\text{rank}(G_2) = 4$ .

1. Firstly, he cuts the ciphertext  $C$  into the following parts

$$C = \underbrace{C_1 C_2 C_3 C_4 C_5}_{C'_1} \mid \underbrace{C_6 C_7 C_8 C_9}_{C'_2} \mid \underbrace{C_{10} C_{11} C_{12} C_{13} C_{14}}_{C'_3},$$

with  $|C'_1| = \text{rank}(G_1) = 5$ ,  $|C'_2| = \text{rank}(G_2) = 4$  and  $|C'_3| = \text{rank}(G_1) = 5$ .

2. Secondly, he calculates the inverse automorphisms for  $g_{1_1}$  and  $g_{2_2}$ . The inverse automorphism  $g_{1_1}^{-1}$  can be explained with Nielsen transformations (applied from the left to the right) as

$$(N1)_1 (N2)_{3.1} (N1)_1 (N1)_4 (N2)_{2.4} (N1)_4 (N1)_2 (N2)_{1.2} (N2)_{4.2} (N1)_5 (N1)_3 \\ (N2)_{5.3} (N1)_5 (N2)_{5.3} (N2)_{1.3} (N1)_2 (N2)_{2.3} (N1)_1 (N1)_3 (N1)_4 (N1)_5,$$

hence it is

$$g_{1_1}^{-1} : G_1 \rightarrow G_1 \\ x_1 \mapsto (x_1 x_4 x_2^{-1} x_1 x_3^{-1})^{-1}, \\ x_2 \mapsto x_2 x_4^{-1} x_1 x_3^{-1}, \\ x_3 \mapsto x_3 x_1^{-1}, \\ x_4 \mapsto x_2 x_4^{-2}, \\ x_5 \mapsto (x_3 x_1^{-1} x_5 x_1 x_3^{-1})^{-1}.$$

The inverse automorphism  $g_{2_2}^{-1}$  can be explained with Nielsen transformations (applied from the left to the right) as

$$(N1)_2 (N2)_{4.2} (N1)_3 (N2)_{1.3} (N1)_3 (N1)_1 [(N2)_{3.1}]^2 (N1)_2 (N1)_4 (N2)_{2.4} \\ [(N2)_{2.1}]^3 (N1)_1 (N1)_2 (N1)_4,$$

hence it is

$$g_{2_2}^{-1} : G_2 \rightarrow G_2 \\ y_1 \mapsto y_1 y_3^{-1}, \\ y_2 \mapsto (y_2^2 y_4^{-1} (y_3 y_1^{-1})^3)^{-1}, \\ y_3 \mapsto y_3^2 y_1^{-1} y_3 y_1^{-1}, \\ y_4 \mapsto y_4 y_2^{-1}.$$

3. He applies  $g_{1_1}^{-1}$  on  $C'_1$ , that is,

$$\left( \left( \frac{453037463005}{2} \quad \frac{-6566656978411}{4} \right), \left( \frac{-515958453260453803}{2} \quad \frac{7478679920196901999}{4} \right), \right. \\ \left. \left( \frac{129695411169}{2} \quad \frac{-187990033891}{2} \right), \left( \frac{12010438543010031}{2} \quad \frac{-174088097591505391}{2} \right), \right. \\ \left. \left( \frac{3968201970233}{2} \quad \frac{-57518027287927}{4} \right), \left( \frac{83406030953}{2} \quad \frac{-1208948133265}{4} \right), \left( \frac{-65207575}{2} \quad \frac{991192539}{4} \right), \right. \\ \left. \left( \frac{529958232109}{2} \quad \frac{-7681602973983}{2} \right), \left( \frac{12234456659}{2} \quad \frac{-177335180327}{2} \right), \left( -4494561 \quad \frac{68319905}{2} \right) \right),$$

and gets

$$\left( \left( \frac{665}{2} \quad \frac{-23229}{4} \right), \left( -57866 \quad \frac{3180525}{2} \right), \left( 15 \quad -109 \right), \left( \frac{109363}{2} \quad \frac{-745561}{4} \right), \left( \frac{729437}{2} \quad \frac{17021361}{4} \right), \right. \\ \left. \left( -29 \quad \frac{1013}{2} \right), \left( 2694 \quad -74036 \right), \left( 4 \quad -29 \right), \left( -4773 \quad \frac{32539}{2} \right), \left( 102117 \quad \frac{2382893}{2} \right) \right).$$

Now, he proves which matrix is an element in  $U' := \{V_1, V_2, \dots, V_{14}\}$ .

It is  $\begin{pmatrix} -57866 & \frac{3180525}{2} \\ 2694 & -74036 \end{pmatrix} \notin U'$ . Hence, Bob knows that this matrix is an ephemeral key from Alice and does not belong to the plaintext.

The other matrices are elements in  $U'$  and with the assignment

$$a_i \hat{=} V_j \iff j \equiv i \pmod{7}$$

he knows

$$\begin{aligned} \begin{pmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{pmatrix} &= V_2 \hat{=} a_2 = \text{I}, \\ \begin{pmatrix} 15 & -109 \\ 4 & -29 \end{pmatrix} &= V_4 \hat{=} a_4 = \text{L}, \\ \begin{pmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{pmatrix} &= V_9 \hat{=} a_2 = \text{I}, \\ \begin{pmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{pmatrix} &= V_{12} \hat{=} a_5 = \text{K}, \end{aligned}$$

analogous for the ciphertext sequences  $C'_2$  and  $C'_3$ .

He applies  $g_2^{-1}$  on  $C'_2$ , that is,

$$\begin{aligned} &\left( \begin{pmatrix} \frac{-1104332496534507861}{2} & \frac{-25304312660337129571}{4} \\ 31604200843034185 & \frac{724168293536436571}{2} \end{pmatrix}, \right. \\ &\left. \begin{pmatrix} -480689945680474129277 & 10323650084255317045974 \\ -143263719821090419728 & 3076836797799093562123 \end{pmatrix}, \right. \\ &\left. \begin{pmatrix} \frac{22386390293811}{2} & \frac{512954405587601}{4} \\ -407276382779 & \frac{-9332197468925}{2} \end{pmatrix}, \right. \\ &\left. \begin{pmatrix} -186180075388817073675749582 & \frac{7997079898367833227219056023}{2} \\ 5914022532266907628279666 & -127013888603589241202576880 \end{pmatrix} \right) \end{aligned}$$

and gets

$$\left( \begin{pmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{pmatrix}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix}, \begin{pmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{pmatrix}, \begin{pmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{pmatrix} \right).$$

Now, he proves which matrix is an element in  $U' := \{V_1, V_2, \dots, V_{14}\}$ .

It is  $\begin{pmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{pmatrix}, \begin{pmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{pmatrix} \notin U'$ . Hence, Bob knows that these matrices are ephemeral keys from Alice and do not belong to the plaintext.

The other matrices are elements in  $U'$  and with the assignment

$$a_i \hat{=} V_j \iff j \equiv i \pmod{7}$$



he knows

$$\begin{pmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{pmatrix} = V_3 \hat{=} a_3 = \mathbf{E},$$

$$\begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix} = V_1 \hat{=} a_1 = \mathbf{B}.$$

He applies  $g_{3_1}^{-1} = g_{1_1}^{-1}$  on  $C'_2$ , that is,

$$\begin{pmatrix} \left( \frac{-1616087435846771117}{2} & \frac{10844781227098250059}{2} \right) \\ \left( 70532776776146599 & \frac{-473311354639843285}{2} \right) \\ \left( \frac{5117735040480436319307}{2} & \frac{-34342644872543531950151}{2} \right) \\ \left( 118098476048874935309 & \frac{-792501759245893528165}{2} \right) \\ \left( \frac{1176330057042013989893}{2} & \frac{-7893782128685642713947}{2} \right) \\ \left( -79397180640094685191 & \frac{532796082062893539917}{2} \right) \\ \left( -3473922528580 & \frac{23311814068153}{2} \right) \\ \left( -110342647234 & 370228071430 \right) \\ \left( 30697842540 & \frac{-205999121749}{2} \right) \\ \left( 9149177258 & -30697963178 \right) \end{pmatrix},$$

and gets

$$\begin{pmatrix} \left( \frac{-843429}{2} & \frac{-19325129}{2} \right) \\ \left( -122869 & \frac{-2815245}{2} \right) \\ \left( \frac{3243}{2} & \frac{-204199}{2} \right) \\ \left( -59 & \frac{3715}{2} \right) \end{pmatrix}, \begin{pmatrix} \left( \frac{621893}{2} & \frac{-34178721}{2} \right) \\ \left( 14351 & \frac{-788719}{2} \right) \\ \left( -71714 & \frac{3080365}{2} \right) \\ \left( 2278 & -48924 \right) \end{pmatrix}, \begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix},$$

Now, he proves which matrix is an element in  $U' := \{V_1, V_2, \dots, V_{14}\}$ .

It is  $\begin{pmatrix} \frac{621893}{2} & \frac{-34178721}{2} \\ 14351 & \frac{-788719}{2} \end{pmatrix}, \begin{pmatrix} \frac{3243}{2} & \frac{-204199}{2} \\ -59 & \frac{3715}{2} \end{pmatrix}, \begin{pmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{pmatrix} \notin U'$ . Hence, Bob knows that these matrices are ephemeral keys from Alice and do not belong to the plaintext.

The other matrices are elements in  $U'$  and with the assignment

$$a_i \hat{=} V_j \iff j \equiv i \pmod{7}$$

he knows

$$\begin{pmatrix} \frac{-843429}{2} & \frac{-19325129}{2} \\ -122869 & \frac{-2815245}{2} \end{pmatrix} = V_{13} \hat{=} a_6 = \mathbf{0},$$

$$\begin{pmatrix} -563 & 1889 \\ 76 & -255 \end{pmatrix} = V_1 \hat{=} a_1 = \mathbf{B}.$$

4. Bob reconstructs the message:

ILIKEBOB

## 9.1. Chosen plaintext attacks on the cryptosystem which uses automorphisms on plaintext sequences

In a chosen plaintext attack (see Section 1.1 or for instance also [BFKR15, Section 3.1]) Eve gives a blackbox, which does the encryption procedure, plaintexts of her choice and gets the corresponding ciphertexts.

Each plaintext is encrypted different also if the automorphisms are the same, because of the ephemeral keys. If, for example, the same letter is given to the blackbox, there are different possibilities to write this letter as a matrix in  $U'$ , because of the agreement

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N},$$

and  $|U'| = k \cdot N$  for a fixed  $k \geq 2$  unknown by Eve. If she gives only one letter to the blackbox, she will get a sequence of matrices of the length  $|G_i|$ , encrypted with an automorphism of the set  $Aut(G_i)$ . The blackbox uses  $|G_i| - 1$  ephemeral matrices for the encryption. The constructive membership problem for the chosen matrices  $U' \in SL(2, \mathbb{Q})$  and the ephemeral keys is unknown and in addition Eve does not know the used ephemeral keys (matrices) for encryption, hence she cannot decide which automorphisms were used for encryption nor how the set  $U'$  look likes.

### Conclusion concerning chosen plaintext attacks

This cryptosystem is secure against chosen plaintext attacks.

## 9.2. Chosen ciphertext attacks on the cryptosystem which uses automorphisms on plaintext sequences

In a chosen ciphertext attack (see Section 1.1 or for instance also [BFKR15, Section 3.1]) Eve chooses ciphertexts and sends these to a blackbox, she then gets the corresponding plaintexts back.

For example Eve gets to some parts of a given ciphertext the corresponding plaintext. These parts can be chosen by her.

Let

$$C = C_1 C_2 \cdots C_{z+\beta}$$

be the ciphertext generated as explained above. In this kind of attacks Eve gets for example the plaintext units  $s_j$  and  $s_{j+1}$ , which are letters of the plaintext alphabet, to the ciphertext matrix  $C_j$  and  $C_{j+1}$ , for a  $j$  with  $1 \leq j \leq z + \beta - 1$ .

This is no help to generate the set  $V'$  or to get hints for the used automorphisms.

### Conclusion concerning chosen ciphertext attacks

This cryptosystem is secure against chosen ciphertext attacks.

## Chapter 10

# Additional cryptographic protocols using automorphisms of finitely generated free groups

This chapter introduces **Protocol 11**, an ElGamal like public key cryptosystem, and **Protocol 12**, a challenge and response system. **Protocol 11** is published in [MR15] and [MR16]. Both systems are based on combinatorial group theory and uses the ideas behind the private key cryptosystems in the previous sections. Thus, they also need a finitely generated free group  $F$ , automorphisms on  $F$  and a faithful representation from  $F$  into  $SL(2, \mathbb{Q})$ .

Firstly, we introduce **Protocol 11**, discuss the security, give proposal for variations and an example. Secondly, **Protocol 12** is described and an example can be found in the appendix.

### 10.1. ElGamal like public key cryptosystem using automorphisms on a finitely generated free group $F$ (Protocol 11)

Now, we describe **Protocol 11**, a public key cryptosystem, for Alice and Bob which is inspired by the ElGamal cryptosystem, see Section 1.2.2 (or for instance [ELG85] or [MSU08, Section 1.3]).

Let  $X = \{x_1, x_2, \dots, x_N\}$ ,  $N \geq 3$ , be the free generating set of the finitely generated free group  $F = \langle X \mid \ \rangle$ . It is  $X^{\pm 1} = X \cup X^{-1}$ . The message is an element  $m \in S^*$ ,  $S^*$  denotes the set of all freely reduced words with letters in  $X^{\pm 1}$ . Public are the free group  $F$ , its free generating set  $X$  and an element  $a \in S^*$ . The automorphism  $f$ , given as a Nielsen transformation or Whitehead-Automorphisms, should be chosen randomly, for example as it is described in Section 4.4.

**Protocol 11**, an ElGamal like public key cryptosystem, with public parameters determined by Alice is now as follows:

**Public parameters:** The finitely generated free group  $F = \langle X \mid \ \rangle$ , a freely reduced word  $a \neq 1$  in the free group  $F$  and an automorphism  $f : F \rightarrow F$  of infinite order.

#### Encryption and Decryption Procedure:

1. Alice chooses privately a natural number  $n$  and publishes the element  $f^n(a) =: c \in S^*$ .
2. Bob picks privately a random  $t \in \mathbb{N}$  and his message  $m \in S^*$ . The number  $t$  is an ephemeral key for this message, he changes  $t$  for each message  $m$ , because of Remark 10.1.1. He calculates the freely reduced elements

$$m \cdot f^t(c) =: c_1 \in S^* \quad \text{and} \quad f^t(a) =: c_2 \in S^*.$$

He sends the ciphertext  $(c_1, c_2) \in S^* \times S^*$  to Alice.

3. Alice calculates

$$\begin{aligned}
 c_1 \cdot (f^n(c_2))^{-1} &= m \cdot f^t(c) \cdot (f^n(c_2))^{-1} \\
 &= m \cdot f^t(f^n(a)) \cdot (f^n(f^t(a)))^{-1} \\
 &= m \cdot f^{t+n}(a) \cdot (f^{n+t}(a))^{-1} \\
 &= m,
 \end{aligned}$$

and gets the message  $m$ .

**Protocol 11**, an ElGamal like public key cryptosystem, is summarized in Table 10.1 (page 236).

Table 10.1.: Summary of **Protocol 11**: ElGamal like public key cryptosystem using automorphisms on a finitely generated free group  $F$

<b>Public Parameters</b>	
Free group $F = \langle X \mid \ \rangle$ , a freely reduced word $a \neq 1$ in $F$ and an automorphism $f : F \rightarrow F$ of infinite order.	
Alice	Bob
Key Creation	
Choose private key $n \in \mathbb{N}$ . Compute $f^n(a) =: c \in S^*.$ ( $S^*$ denotes the set of all freely reduced words with letters in $X^{\pm 1}$ .) Publish $c$ .	
Encryption	
	Choose plaintext $m \in S^*$ . Choose random ephemeral key $t \in \mathbb{N}$ . Compute $m \cdot f^t(c) =: c_1 \in S^* \quad \text{and} \quad f^t(a) =: c_2 \in S^*.$ Send ciphertext $(c_1, c_2) \in S^* \times S^*$ to Alice. $\xleftarrow{(c_1, c_2)}$
Decryption	
Compute $  \begin{aligned}  c_1 \cdot (f^n(c_2))^{-1} &= m \cdot f^t(c) \cdot (f^n(c_2))^{-1} \\  &= m \cdot f^t(f^n(a)) \cdot (f^n(f^t(a)))^{-1} \\  &= m \cdot f^{t+n}(a) \cdot (f^{n+t}(a))^{-1} \\  &= m,  \end{aligned}  $ which is the message from Bob.	

**Remark 10.1.1.** It is important that different random ephemeral keys  $t$  are used to encrypt different messages. As it is for the standard ElGamal cryptosystem (see [MvOV97]). Suppose that Bob uses the same ephemeral key  $t$  to encrypt two messages  $m_1$  and  $m_2$  and assume that  $m_1$  is known. The ciphertext pairs are  $(c_1, c_2)$  and  $(c'_1, c'_2)$ , with  $c_2 = c'_2$ ,  $c_1 = m_1 \cdot f^t(c)$  and  $c'_1 = m_2 \cdot f^t(c)$ . Eve only has to calculate  $c'_1 \cdot (c_1)^{-1} \cdot m_1$  to get the message  $m_2$ .

**Security 10.1.2.** A possible attacker, Eve, can see the elements  $c, c_1, c_2 \in S^*$ . She does not know the free length of  $m$  and the cancellations between  $m$  and  $f^t(c)$  in  $c_1$ . It could be possible that  $m$  is completely canceled by the first letters of  $f^t(c)$ . Hence, she cannot determine  $m$  from the given  $c_1$ . Eve just sees words,  $f^t(a)$  and  $f^n(a)$ , in the free generating set  $X$  from which it is unlikely to realize the exponents  $n$  and  $t$ , that is, the private keys from Alice and Bob, respectively. The security is based on the Diffie-Hellman problem and discrete logarithm problem in cyclic subgroups of automorphisms on finitely generated free groups.

**Variation 10.1.3.** We give some ideas to enhance the security, they can also be combined:

1. The element  $a \in S^*$  could be taken as a common private secret between Alice and Bob. They could use for example the Anshel-Anshel-Goldfeld key exchange protocol (see [MSU08]) to agree on the element  $a$ .
2. Alice and Bob agree on a faithful representation from  $F$  into the special linear group of all  $2 \times 2$  matrices with entries in  $\mathbb{Q}$ , that is,  $g : F \rightarrow \text{SL}(2, \mathbb{Q})$ . Now,  $m \in S^*$  and Bob sends  $g(m) \cdot g(f^t(c)) =: c_1 \in \text{SL}(2, \mathbb{Q})$  instead of  $m \cdot f^t(c) =: c_1 \in S^*$ ;  $c$  and  $c_2$  remain the same. Therefore, Alice calculates  $c_1 \cdot (g(f^n(c_2)))^{-1} = g(m)$  and hence the message  $m = g^{-1}(g(m)) \in S^*$ . This variation in addition extends the security certification to the constructive membership problem in the matrix group  $\text{SL}(2, \mathbb{Q})$  (see [EKL14]).

We now explain this variation in more details.

In addition to  $X = \{x_1, x_2, \dots, x_N\}$  Alice chooses a second abstract set  $Y = \{y_1, y_2, \dots, y_N\}$ ,  $X \cap Y = \emptyset$ , which generates a free group  $F' = \langle Y \mid \ \rangle$  of rank  $N$ . The automorphism  $f$  from Alice is an automorphism on a free group of rank  $|X|$  if we identify  $x_i$  with  $y_i$  for  $i = 1, 2, \dots, N$ , then  $f$  is also an automorphism of  $F'$ , because  $|X| = |Y|$  and hence  $F'$  is isomorphic to  $F$ , see Theorem 4.3.7.

Alice needs a faithful representation of  $\langle X \cup Y \mid \ \rangle$  into  $\text{SL}(2, \mathbb{Q})$ , such that

$$\begin{aligned}
 g : \langle X \cup Y \mid \ \rangle &\rightarrow \text{SL}(2, \mathbb{Q}) \\
 x_i &\mapsto M_i, && \text{with } i = 1, 2, \dots, N \text{ and } M_i \in \text{SL}(2, \mathbb{Z}), \\
 y_i &\mapsto W_i, && \text{with } i = 1, 2, \dots, N \text{ and } W_i \in \text{SL}(2, \mathbb{Q}) \text{ and } W_i \notin \text{SL}(2, \mathbb{Z}).
 \end{aligned}$$

Thus, each  $W_i$  has at least one entry which is an element in  $\mathbb{Q} \setminus \mathbb{Z}$ . The set  $g(X) \cup g(Y)$  is a free generating set of a free subgroup in  $\text{SL}(2, \mathbb{Q})$  of rank  $2N$ .

- a) The public element from Alice is as before  $c = f^n(a) \in S^*$ , with private key  $n \in \mathbb{N}$ .
- b) Bob chooses privately a message  $m \in S^*$ , a random  $t \in \mathbb{N}$  and calculates the element  $c_2 = f^t(a) \in S^*$  as before. After this he computes  $f^t(c) = f^t(f^n(a)) = f^{t+n}(a) \in S^*$  and writes it as a word in  $Y$  whereby he used the assignment  $x_i \mapsto y_i$  for  $1 \leq i \leq N$ . We denote  $f^t(c)$  as  $f_Y^t(c)$  when  $f^t(c)$  is written as a word in  $Y$ . The element  $f_Y^t(c)$  is a reduced word in  $Y$ . Bob's element  $c_1 = m \cdot f_Y^t(c)$  is now a reduced word in  $X \cup Y$ .

He applies the faithful representation  $g$  on this element. It is

$$g(m \cdot f_Y^t(c)) = \underbrace{g(m)}_{\in \text{SL}(2, \mathbb{Z})} \cdot \underbrace{g(f_Y^t(c))}_{\in \text{SL}(2, \mathbb{Q})} =: c'_1 \in \text{SL}(2, \mathbb{Q}).$$

Instead of  $(c_2, c_1) \in S^* \times S^*$  he sends  $(c_2, c'_1) \in S^* \times \text{SL}(2, \mathbb{Q})$  to Alice.

c) Firstly, Alice calculates  $f^n(c_2)$  and hence gets the same element  $f^t(c)$  as Bob, because

$$f^n(c_2) = f^n(f^t(a)) = f^{n+t}(a) = f^{t+n}(a) = f^t(f^n(a)) = f^t(c).$$

Secondly, she writes  $f^n(c_2)$  as a word in  $Y$ , thus she gets  $f_Y^t(c)$ . Thirdly, she uses the faithful representation  $g$  to calculate  $g(f_Y^t(c))$  and together with  $c'_1$  she gets

$$c'_1 \cdot (g(f_Y^t(c)))^{-1} = g(m) \cdot g(f_Y^t(c))(g(f_Y^t(c)))^{-1} = g(m) \in \text{SL}(2, \mathbb{Z}).$$

She gets a matrix in  $\text{SL}(2, \mathbb{Z})$  and she knows that this matrix is a word in the letters of  $M_i$ ,  $1 \leq i \leq N$ , hence there is an algorithm (see Remark 4.3.12 and Theorem 4.3.13) to write  $g(m)$  as a word in  $g(X)$  and therefore as a word in  $X$ . Thus, she is able to reconstruct  $m$ .

An eavesdropper, Eve, gets a matrix  $c'_1 \in \text{SL}(2, \mathbb{Q})$  and she is not able to write it as a word in  $X \cup Y$  (because there is no algorithm known to solve the constructive membership problem in a (discrete) free subgroup of  $\text{SL}(2, \mathbb{Q})$  of rank greater than or equal to 2 ([EKL14]), which is not in  $\text{SL}(2, \mathbb{Z})$ ). Thus, she cannot get the situation as in the cryptosystem without the faithful representation  $g$  into  $\text{SL}(2, \mathbb{Q})$ . There is no hint for the message  $m$ , instead of the system above in which it is possible that an initial segment of  $m$  is visible whereby Eve does not know how long this initial segment is and if it is relay visible. Thus, this variation extends the security certification to the constructive membership problem in the matrix group  $\text{SL}(2, \mathbb{Q})$ .

#### Example 10.1.4.

This example, see also Appendix C.11, is a very small one and it is just given for illustration purposes. Bob wants to send a message to Alice.

The **public parameters** are the free group  $F$  of rank 3 with free generating set  $X = \{x, y, z\}$ , the freely reduced word  $a \in F$ , with  $a := x^2yz^{-2}y$  and the automorphism  $f : F \rightarrow F$ , which is given, for this example, by the regular Nielsen transformation:  $[(N2)_{1,2}]^2 (N2)_{3,2} (N1)_3 (N2)_{2,3}$ , thus, it is:

$$\begin{aligned} f : F &\rightarrow F \\ x &\mapsto xy^2, \\ y &\mapsto z^{-1}, \\ z &\mapsto y^{-1}z^{-1}. \end{aligned}$$

1. Alice's private key is  $n = 7$ . Thus, she gets the automorphism

$$\begin{aligned} f^7 : F &\rightarrow F \\ x &\mapsto xy^2z^{-1}y(yz)^2(zyz^2y)^2zy, \\ y &\mapsto y^{-1}((z^{-1}y^{-1}z^{-1})^2y^{-1}z^{-1})^2z^{-1}y^{-1}z^{-2}, \\ z &\mapsto (((y^{-1}z^{-1})^2z^{-1})^2y^{-1}z^{-2})^2y^{-1}(z^{-1}y^{-1}z^{-1})^2z^{-1}. \end{aligned}$$

Her public key is

$$c := f^7(a) = (xy^2z^{-1}y(yz)^2(zyz^2y)^2zy)^2(z^2y)^2 \\ ((zyz)^2yz)^2zyz^2yz^{-1}.$$

2. Bob privately picks the ephemeral key  $t = 5$  and gets the automorphism

$$f^5 : F \rightarrow F \\ x \mapsto xy^2z^{-1}y^2z(zy)^2, \\ y \mapsto y^{-1}(z^{-1}y^{-1}z^{-1})^2z^{-1}, \\ z \mapsto ((y^{-1}z^{-1})^2z^{-1})^2y^{-1}z^{-2}.$$

His message for Alice is  $m = z^{-2}y^2zx^2y^{-1}x^{-1}$ . He calculates

$$c_1 = m \cdot f^5(c) \\ = z^{-2}y^2zx^2(yz^{-1})^2((z^{-1}y^{-1}z^{-2}y^{-1})^2z^{-2}y^{-1})^2(z^{-1}y^{-1}z^{-1})^2z^{-1}y^{-1} \\ (((z^{-1}y^{-1}z^{-1})^2y^{-1}z^{-1})^2z^{-1}y^{-1}z^{-1}y^{-1}z^{-1})^2(z^{-1}y^{-1}z^{-2}y^{-1})^2z^{-1} \\ y^{-1}z^{-1})^2((z^{-1}y^{-1}z^{-2}y^{-1})^2z^{-2}y^{-1})^2(z^{-1}y^{-1}z^{-1})^2z^{-1}xy^2z^{-1}y(z^{-1} \\ (((z^{-1}y^{-1}z^{-2}y^{-1})^2z^{-2}y^{-1})^2(z^{-1}y^{-1}z^{-1})^2z^{-1}y^{-1})^3(z^{-1}y^{-1}z^{-1})^2 \\ y^{-1}z^{-1}((z^{-1}y^{-1}z^{-2}y^{-1})^2z^{-2}y^{-1})^2(z^{-1}y^{-1}z^{-1})^2y^{-1})^3z^{-1} \\ ((z^{-1}y^{-1}z^{-2}y^{-1})^2z^{-2}y^{-1})^2(z^{-1}y^{-1}z^{-1})^2y^{-1}z^{-1}y$$

and

$$c_2 := f^5(a) = (xy^2z^{-1}y^2z(zy)^2)^2z^2y(zyz)^2zyz^{-1}.$$

The ciphertext for Alice is the tuple  $(c_1, c_2)$ .

3. Alice first computes

$$(f^7(c_2))^{-1} = y^{-1}((((zy)^2z)^2zyz)^2zy(zyz)^2zy \\ ((zyz)^2yz)^2zyz)^2zy(((zyz)^2yz)^2zyzyz)^2 \\ (zyz^2y)^2z)^2y((((z^2y)^2zy)^2z^2yzy)^2z \\ (zyz^2y)^2zy)^2z(zyz^2y)^2z^2y \\ (((zyz)^2yz)^2zyzyz)^2(zyz^2y)^2z \\ (zy^{-1})^2y^{-1}x^{-1})^2$$

and gets  $m$  by

$$m = c_1 \cdot (f^7(c_2))^{-1} = z^{-2}y^2zx^2y^{-1}x^{-1}.$$

## 10.2. Challenge and response protocol using automorphisms on a finitely generated free group $F$ (Protocol 12)

We use the idea behind the public key cryptosystem based on Nielsen transformations in the previous section to develop a challenge and response protocol. More precisely this is a symmetric

key authentication protocol (see for example [BBFT10] or [BNS10, Section 18.3]).

First we start with a general outline of this challenge and response system. The structure is adapted on a model which is now used for most password and password back-up schemes, see [BBFT10, p. 6]. Afterwards we make suggestions for possible challenges and give a security analysis.

General outline of this symmetric key authentication protocol:

In this variation each prover is assigned to an automorphism  $f$  of infinite order (Nielsen transformation or Whitehead-Automorphisms) of a free group  $F$  with rank  $N \geq 3$ , that is,  $F = \langle X \mid \ \rangle$  with  $X = \{x_1, x_2, \dots, x_N\}$ . Due to this, the common shared secret between the prover and the verifier is the tuple  $(P, f)$  with  $P$  a standard password for the prover and  $f$  the associated challenge automorphism.

This is a symmetric key cryptographic authentication protocol, thus, both the prover and verifier use a single common private key within the authentication process, which is here  $f$ .

1. The prover and verifier communicate directly, either face-to-face or by a public key method, to setup a common shared secret  $(P, f)$  with  $P$  a standard password and  $f$  the challenge automorphism of a free group of rank  $N$ . Each prover's challenge automorphism is unique to that prover. The password is chosen by the prover while the challenge automorphism is randomly chosen.
2. The prover presents the password to the verifier. The verifier presents a "question" (see possible challenges for the prover below). The assumption is that this "question" is difficult in the sense that it is infeasible to answer it if the automorphism  $f$  is unknown. This is repeated a finite number of times. If all answers are correct the prover (and the password) is verified.
3. The cryptographic protocol is then repeated from the viewpoint of the prover, authenticating the verifier to the prover.

We give examples for questions which are very unlikely to answer correctly if the challenge automorphism is unknown.

Possible challenges for the prover:

We propose four types of questions for the challenges.

1. What is the matrix  $M = \varphi(f^n(w))$ , given  $w \in F$  (a freely reduced word in  $F$ ),  $n \in \mathbb{N}$  and a faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ ? The verifier takes care that each matrix  $\varphi(x_i)$ ,  $1 \leq i \leq N$ , has at least one entry in  $\mathbb{Q} \setminus \mathbb{Z}$ .
2. What is the trace of the matrix  $M = \varphi(f^n(w))$ , given  $w \in F$  (a freely reduced word in  $F$ ),  $n \in \mathbb{N}$  and a faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ ? The verifier takes care that each matrix  $\varphi(x_i)$ ,  $1 \leq i \leq N$ , has at least one entry in  $\mathbb{Q} \setminus \mathbb{Z}$ .
3. What is the entry  $M_{x,y}$  of the matrix  $M = \varphi(f^n(w))$ , given  $w \in F$  (a freely reduced word in  $F$ ),  $n \in \mathbb{N}$  and a faithful representation  $\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})$ , with  $x, y \in \{1, 2\}$  and  $x$  gives the row and  $y$  the column in the matrix  $M$ ? A variation could be given if the entry  $M_{x,y}$  is an integer, then it could be ask for certain digits of an entry  $M_{x,y}$ , for example for the last 7 digits.



10.2. Challenge and response protocol using automorphisms on a finitely generated free group  $F$  (Protocol 12)

4. Questions as in 1. and 2. but the faithful representation  $\varphi$  could be public or also a part of the common shared secret between the verifier and the prover.

**Protocol 12** is summarized in Table 10.2 (page 241) with a challenge of kind 1.

Table 10.2.: Summary of **Protocol 12**: Challenge and response protocol using automorphisms on finitely generated free groups

<b>Private Parameters</b>	
Free group $F$ with free generating set $X = \{x_1, x_2, \dots, x_N\}$ , $N \geq 3$ ; an automorphism $f \in \text{Aut}(F)$ of infinite order and a common password $P$ . The shared secret is the tuple $(P, f)$ .	
<b>Verifier</b>	<b>Prover</b>
	Present the password $P$ to the verifier $\xrightarrow{P}$
Take challenge automorphism $f$ corresponding to password $P$ . Choose <ul style="list-style-type: none"> <li>• a faithful representation <math>\varphi : F \rightarrow \text{SL}(2, \mathbb{Q})</math>; take care that each matrix <math>\varphi(x_i)</math>, <math>1 \leq i \leq N</math>, has at least one entry in <math>\mathbb{Q} \setminus \mathbb{Z}</math>;</li> <li>• a freely reduced word <math>w \in F</math>;</li> <li>• <math>n \in \mathbb{N}</math>.</li> </ul>	Challenge: $(\varphi, w, n)$ $\xrightarrow{\hspace{10em}}$
Compute $M' = \varphi(f^n(w)).$	Compute the response $M$ and send it to the verifier $M = \varphi(f^n(w)).$
	Response: $M$ $\xleftarrow{M}$
Proof if $M' = M$ .	

**Security 10.2.1.** For security analysis we assume that an adversary or eavesdropper has access to the encrypted form of the transmission but is passive in that the adversary will not change any transmissions.

An eavesdropper, Eve, is interested in the challenge automorphism  $f$ . In the first variation Eve gets a faithful representation  $\varphi$ , an element  $w \in F$ , a natural number  $n$  and the matrix  $M$  or the trace of  $M$ .

If the trace is given it is very unlikely that she reconstructs the correct matrix  $M$ . If she gets the matrix  $M$  Eve is not able to write  $M$  as a word in  $\varphi(x_i)$ , because it is no algorithm known to solve the constructive membership problem in  $\text{SL}(2, \mathbb{Q})$  for (discrete) free subgroups of rank greater than or equal to 2, see Remark 4.3.14. In the second variation an eavesdropper gets  $w \in F$ , the natural number  $n$ , the Matrix  $M$  or the trace of  $M$  and for each challenge another faithful representation  $\varphi$ . Now, an eavesdropper has the problem, that the faithful representation  $\varphi$  is changed at every challenge.

Furthermore, if Eve stores the challenges and responses she cannot use them to pose as the prover, because  $\varphi(f^n(w))\varphi(f^b(w)) = \varphi(f^n(w)f^b(w)) \neq \varphi(f^{n+b}(w))$  because  $f^{n+b}(w) = f^b(f^n(w))$  and in general  $f^b(f^n(w)) \neq f^n(w)f^b(w)$  beside this, also the word  $w$  can be changed at every challenge.

There are infinitely many possibilities for the word  $w \in F$ , the number  $n \in \mathbb{N}$  and also the faithful representation  $\varphi$  into  $\mathrm{SL}(2, \mathbb{Q})$ . Thus, each challenge is used only once. Therefore, replay attacks, in which an eavesdropper records a communication session and replays parts of the session or the whole session (see [MvOV97]) is avoided. If a challenge is used twice the verifier or the prover, respectively, knows that it is an attack.

**Example 10.2.2.** An example for one challenge and the corresponding response together with the GAP-Code and Maple-Code can be found in Appendix C.12.

# Appendix A

## Additional definitions

### A.1. Boolean formulae

This appendix-section is based on the books [Lig06], [CK02] and [Weg87], it reminds the reader of the theory of boolean formulae and gives the required definitions for Section 5.3.2.

**Definition A.1.1.** A **boolean function** in the (boolean) variables  $\psi_1, \psi_2, \dots, \psi_n$  is a map  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ .

Some examples of boolean functions include

- the 0-ary constant functions 0 and 1,
- the unary function  $\neg$  (negation),
- the binary functions  $\vee$  (OR),  $\wedge$  (AND),  $\oplus$  (EXCLUSIVE OR),  $\Rightarrow$  (implication, where  $\psi_1 \Rightarrow \psi_2$  is defined by  $\neg\psi_1 \vee \psi_2$ ) and  $\equiv$  (also  $\Leftrightarrow$ ) (equivalence, also called bimplication).

**Definition A.1.2.** A **literal** is a boolean variable  $\psi_i$  or its negation  $\neg\psi_i$ .

The negation of a literal  $\neg\psi_i$  is  $\psi_i$ .

**Definition A.1.3.** A **boolean formula** or just formula (over the De Morgan basis  $\{\wedge, \vee, \oplus, 0, 1\}$ ) is defined recursively as follows:

- The constants (that is 0-place connectives) 0 (FALSE) and 1 (TRUE) are boolean formulae.
- A boolean variable is a boolean formula.
- If  $\Psi_1$  and  $\Psi_2$  are boolean formulae, then  $\neg\Psi_1$  (**negation**),  $(\Psi_1 \wedge \Psi_2)$  (**conjunction**) and  $(\Psi_1 \vee \Psi_2)$  (**disjunction**) are boolean formulae.

The connectives  $\wedge$  and  $\vee$  associate to the right, so that  $\Psi_1 \vee \Psi_2 \vee \Psi_3$  means  $\Psi_1 \vee (\Psi_2 \vee \Psi_3)$  and  $\Psi_1 \wedge \Psi_2 \wedge \Psi_3$  means  $\Psi_1 \wedge (\Psi_2 \wedge \Psi_3)$ .

**Definition A.1.4.** Let  $\psi^1 = \psi$ ,  $\psi^0 = \neg\psi$  and let  $\psi_1, \psi_2, \dots, \psi_n$  be some distinct literals. A conjunction of the form

$$\psi_1^{a_1} \wedge \psi_2^{a_2} \wedge \dots \wedge \psi_n^{a_n},$$

with  $a_i \in \{0, 1\}$ ,  $1 \leq i \leq n$ , is called a **term** (or minterm or simple formula).

**Definition A.1.5.** Let  $\psi^1 = \psi$ ,  $\psi^0 = \neg\psi$  and let  $\psi_1, \psi_2, \dots, \psi_n$  be some distinct literals. A disjunction of the form

$$\psi_1^{a_1} \vee \psi_2^{a_2} \vee \dots \vee \psi_n^{a_n},$$

with  $a_i \in \{0, 1\}$ ,  $1 \leq i \leq n$ , is called a **clause** (or maxterm).

**Definition A.1.6.** A formula is in **Conjunctive Normal Form (CNF)** if it can be presented as

$$\Psi = \psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_n,$$

where  $\psi_1, \psi_2, \dots, \psi_n$  denote any clauses. (In words: Any formula in CNF is a conjunction of disjunctions of literals.)

**Definition A.1.7.** A formula is in **Disjunctive Normal Form (DNF)** if it can be presented as

$$\Psi = \psi_1 \vee \psi_2 \vee \cdots \vee \psi_n,$$

where  $\psi_1, \psi_2, \dots, \psi_n$  denote any terms. (In words: Any formula in DNF is a disjunction of conjunctions of literals.)

As may be observed, for a particular formula there may exist many different CNF which are equivalent.

**Definition A.1.8.** A formula

$$\Psi = \psi_1 \vee \psi_2 \vee \cdots \vee \psi_n$$

is in **minimal DNF form** if and only if there does not exist a logically equivalent formula in DNF composed of  $m$  terms where  $m < n$ .

**Definition A.1.9.** A formula

$$\Psi = \psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_n$$

is in **minimal CNF form** if and only if there does not exist a logically equivalent formula in CNF composed of  $m$  clauses where  $m < n$ .

Any formula can be transformed into a logically equivalent CNF or DNF form. It can be noted that in general case the minimal DNF of a formula is not defined in a unique way.

**Remark A.1.10.** It is known, that monotone formulae contain only AND ( $\wedge$ ) and OR ( $\vee$ ) operators (but no NEGATIONS ( $\neg$ )).

## A.2. Elementary free groups

To explain elementary free groups we need the first-order theory, because non-free groups that have exactly the same first-order theory as the class of nonabelian free groups are called elementary free groups (or elementarily free groups).

Therefore, we give a brief introduction into first order theory, which is from the book [FGMRS14].

The starting point is a first-order language with equality (always interpreted as the identity relation) containing a binary operation symbol “ $\cdot$ ” (often suppressed in favor of juxtaposition), unary operation symbol “ $^{-1}$ ”, and a constant symbol “ $1$ ”. In particular, this is what it means for  $L_0$  to be appropriate for group theory. A **formula** in this language is a logical expression containing a string of variables  $\bar{x} = (x_1, x_2, \dots, x_n)$ , the logical connectives  $\vee, \wedge, \sim$ , and the quantifiers  $\forall, \exists$ . Here  $\vee$  stands for the disjunction of two propositions,  $\wedge$  for the conjunction of two propositions and  $\sim$  for the negation.

A variable in a formula is called **bound** (or occurs bound) if it is restricted by a quantifier ( $\forall, \exists$ ). Otherwise, the variable is called **free** (or occurs free). A **sentence** is a formula of  $L_0$  in which all variables are bound, or in other words there are no free occurrences of any variable.

A **universal sentence** of  $L_0$  is one of the form  $\forall \bar{x} \{ \phi(\bar{x}) \}$  where  $\bar{x}$  is a tuple of distinct variables,  $\phi(\bar{x})$  is a formula of  $L_0$  containing no quantifiers and containing at most the variables of  $\bar{x}$ . Similarly an **existential sentence** is one of the form  $\exists \bar{x} \{ \phi(\bar{x}) \}$  where  $\bar{x}$  and  $\phi(\bar{x})$  are as above.

**Example A.2.1.** 1. The sentence

$$\forall(x, y) \{ xy = yx \}$$

is a universal sentence describing an abelian group.

2. The sentence

$$\exists(x, y) \{ xy \neq yx \}$$

is an existential sentence describing a nonabelian group.

A **universal-existential sentence** is one of the form  $\forall \bar{x} \exists \bar{y} \{ \phi(\bar{x}, \bar{y}) \}$ . Similarly defined is an **existential-universal sentence**, which is of the form  $\exists \bar{x} \forall \bar{y} \{ \phi(\bar{x}, \bar{y}) \}$ . It is known that every sentence of  $L_0$  is logically equivalent to one of the form  $Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \{ \phi(\bar{x}) \}$ , where  $\bar{x} = (x_1, x_2, \dots, x_n)$  is a tuple of distinct variables, each  $Q_i$  for  $i = 1, 2, \dots, n$  is a quantifier, either  $\forall$  or  $\exists$ , and  $\phi(\bar{x})$  is a formula of  $L_0$  containing no quantifiers and containing freely at most the variables  $x_1, x_2, \dots, x_n$ . Further vacuous quantifications are permitted. Finally a **positive sentence** is one logically equivalent to a sentence constructed using (at most) the connectives  $\vee, \wedge, \forall, \exists$ , that is, no negations are allowed.

If  $G$  is a group then the **universal theory** of  $G$  consists of the set of all universal sentences of  $L_0$  true in  $G$ . Since any universal sentence is equivalent to the negation of an existential sentence it follows that two groups have the same universal theory if and only if they have the same **existential theory**. The set of all sentences of  $L_0$  true in  $G$  is called the **first-order theory** or the **elementary theory** of  $G$ .

The primary examples of elementary free groups are the orientable surface groups  $S_g$  of genus  $g \geq 2$  and the non-orientable surface groups  $N_g$  of genus  $g \geq 4$ .

For more information about elementary free groups see for instance the book [FGMRS14].



## Appendix B

### Additional examples

If there are Nielsen transformations of type  $(N1)$  one after another we can apply them in one step. For example if the Nielsen transformations  $(N1)_5 (N1)_2 (N1)_1 (N2)_{3.2}$  are applied to a set  $(a, b, c, d, e)$  we write instead of

$$\begin{aligned} (a, b, c, d, e) &\xrightarrow{(N1)_5} (a, b, c, d, e^{-1}) \\ &\xrightarrow{(N1)_2} (a, b^{-1}, c, d, e^{-1}) \\ &\xrightarrow{(N1)_1} (a^{-1}, b^{-1}, c, d, e^{-1}) \\ &\xrightarrow{(N2)_{3.2}} (a^{-1}, b^{-1}, cb^{-1}, d, e^{-1}) \end{aligned}$$

the following

$$\begin{aligned} (a, b, c, d, e) &\xrightarrow{(N1)_5(N1)_2(N1)_1} (a^{-1}, b^{-1}, c, d, e^{-1}) \\ &\xrightarrow{(N2)_{3.2}} (a^{-1}, b^{-1}, cb^{-1}, d, e^{-1}). \end{aligned}$$

We use the FGA<sup>1</sup> package in GAP<sup>2</sup>.

#### B.1. Example for automorphisms for Remark 7.0.10

Let  $F$  be a free group of rank 6 with free generating set  $X = \{x_1, x_2, \dots, x_6\}$ . Let  $U$  be a subset of  $F$ , which is Nielsen reduced. It is  $U = \{u_1, u_2, u_3, u_4\}$ , with  $u_1 = x_3^2 x_2^{-1}$ ,  $u_2 = x_1^{-1} x_3 x_5 x_6^{-1}$ ,  $u_3 = x_4 x_2^2 x_3^2$ ,  $u_4 = (x_4^{-1} x_1^{-1} x_3 x_5)^4$ .

In GAP it is

```
LoadPackage("FGA");;
F:=FreeGroup("x1", "x2", "x3", "x4", "x5", "x6");;
AssignGeneratorVariables(F);;
```

and with the operation

```
▷ FreeGeneratorsOfGroup(G)
```

which returns a list of free Nielsen reduced generators, which defines a Nielsen reduced set of the finitely generated subgroup  $G$  of a free group, we can prove that

$$U = \{x_3^2 x_2^{-1}, x_1^{-1} x_3 x_5 x_6^{-1}, x_4 x_2^2 x_3^2, (x_4^{-1} x_1^{-1} x_3 x_5)^4\}$$

<sup>1</sup>Free Group Algorithms. A GAP4 Package by Christian Sievers, TU Braunschweig.

<sup>2</sup>Groups, Algorithms and Programming [GAP15]

is Nielsen reduced:

```
G:=Group(x3^2*x2^-1,x1^-1*x3*x5*x6^-1,x4*x2^2*x3^2,(x4^-1*x1^-1*x3*x5)^4);;
```

```
gap> FreeGeneratorsOfGroup(G);
```

```
[ x3^2*x2^-1, x1^-1*x3*x5*x6^-1, x4*x2^2*x3^2, (x5^-1*x3^-1*x1*x4)^4 ]
```

GAP gives the Nielsen reduced set  $U_{GAP} = \{x_3^2x_2^{-1}, x_1^{-1}x_3x_5x_6^{-1}, x_4x_2^2x_3^2, (x_5^{-1}x_3^{-1}x_1x_4)^4\}$ . This differs to  $U$  only in the last element, which is the inverse of  $u_4$ . To prove that  $U$  is Nielsen reduced the Lemma 4.2.15 must be proved only for  $u_4$  in the set  $U$ , because of  $U_{GAP}$  and the results of GAP. Now, after Lemma 4.2.15 the set  $U$  is Nielsen reduced.

The plaintext from Alice is

$$S = a_4a_2a_1a_2.$$

For encryption she uses the following automorphisms, which are describable with the given Nielsen transformations:

- Automorphism  $f_{x_1}$ :

$$\begin{aligned}
 (x_1, x_2, x_3, x_4, x_5, x_6) &\xrightarrow{(N2)_{5.6}} (x_1, x_2, x_3, x_4, x_5x_6, x_6) \\
 &\xrightarrow{(N1)_4(N1)_5} (x_1, x_2, x_3, x_4^{-1}, x_6^{-1}x_5^{-1}, x_6) \\
 &\xrightarrow{(N2)_{5.4}} (x_1, x_2, x_3, x_4^{-1}, x_6^{-1}x_5^{-1}x_4^{-1}, x_6) \\
 &\xrightarrow{(N1)_3} (x_1, x_2, x_3^{-1}, x_4^{-1}, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N2)_{4.3}} (x_1, x_2, x_3^{-1}, x_4^{-1}x_3^{-1}, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N1)_4} (x_1, x_2, x_3^{-1}, x_3x_4, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N2)_{1.2}} (x_1x_2, x_2, x_3^{-1}, x_3x_4, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N1)_1} (x_2^{-1}x_1^{-1}, x_2, x_3^{-1}, x_3x_4, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N2)_{3.1}} (x_2^{-1}x_1^{-1}, x_2, x_3^{-1}x_2^{-1}x_1^{-1}, x_3x_4, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N1)_3(N1)_1} (x_1x_2, x_2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N2)_{2.6}} (x_1x_2, x_2x_6, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6) \\
 &\xrightarrow{(N2)_{6.1}} (x_1x_2, x_2x_6, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6x_1x_2)
 \end{aligned}$$

Hence, it is

$$\begin{aligned}
 f_{x_1} : F &\rightarrow F \\
 x_1 &\mapsto x_1x_2, \\
 x_2 &\mapsto x_2x_6, \\
 x_3 &\mapsto x_1x_2x_3, \\
 x_4 &\mapsto x_3x_4, \\
 x_5 &\mapsto x_4x_5x_6, \\
 x_6 &\mapsto x_6x_1x_2.
 \end{aligned}$$



- Automorphism  $f_{x_2}$ :

$$\begin{aligned}
 (x_1, x_2, x_3, x_4, x_5, x_6) &\xrightarrow{(N2)_{4.2}} (x_1, x_2, x_3, x_4x_2, x_5, x_6) \\
 &\xrightarrow{(N2)_{6.2}} (x_1, x_2, x_3, x_4x_2, x_5, x_6x_2) \\
 &\xrightarrow{(N2)_{5.6}} (x_1, x_2, x_3, x_4x_2, x_5x_6x_2, x_6x_2) \\
 &\xrightarrow{(N2)_{1.3}} (x_1x_3, x_2, x_3, x_4x_2, x_5x_6x_2, x_6x_2) \\
 &\xrightarrow{(N1)_1(N1)_3} (x_3^{-1}x_1^{-1}, x_2, x_3^{-1}, x_4x_2, x_5x_6x_2, x_6x_2) \\
 &\xrightarrow{(N2)_{3.1}} (x_3^{-1}x_1^{-1}, x_2, x_3^{-2}x_1^{-1}, x_4x_2, x_5x_6x_2, x_6x_2) \\
 &\xrightarrow{(N1)_3(N1)_1} (x_1x_3, x_2, x_1x_3^2, x_4x_2, x_5x_6x_2, x_6x_2) \\
 &\xrightarrow{(N2)_{2.1}} (x_1x_3, x_2x_1x_3, x_1x_3^2, x_4x_2, x_5x_6x_2, x_6x_2)
 \end{aligned}$$

Hence, it is

$$\begin{aligned}
 f_{x_2} : F &\rightarrow F \\
 x_1 &\mapsto x_1x_3, \\
 x_2 &\mapsto x_2x_1x_3, \\
 x_3 &\mapsto x_1x_3^2, \\
 x_4 &\mapsto x_4x_2, \\
 x_5 &\mapsto x_5x_6x_2, \\
 x_6 &\mapsto x_6x_2.
 \end{aligned}$$

- Automorphism  $f_{x_3}$

$$\begin{aligned}
 (x_1, x_2, x_3, x_4, x_5, x_6) &\xrightarrow{(N2)_{1.2}} (x_1x_2, x_2, x_3, x_4, x_5, x_6) \\
 &\xrightarrow{(N2)_{2.3}} (x_1x_2, x_2x_3, x_3, x_4, x_5, x_6) \\
 &\xrightarrow{(N2)_{3.2}} (x_1x_2, x_2x_3, x_3x_2x_3, x_4, x_5, x_6) \\
 &\xrightarrow{(N2)_{4.5}} (x_1x_2, x_2x_3, x_3x_2x_3, x_4x_5, x_5, x_6) \\
 &\xrightarrow{(N2)_{5.2}} (x_1x_2, x_2x_3, x_3x_2x_3, x_4x_5, x_5x_2x_3, x_6) \\
 &\xrightarrow{(N1)_1} (x_2^{-1}x_1^{-1}, x_2x_3, x_3x_2x_3, x_4x_5, x_5x_2x_3, x_6) \\
 &\xrightarrow{(N2)_{6.1}} (x_2^{-1}x_1^{-1}, x_2x_3, x_3x_2x_3, x_4x_5, x_5x_2x_3, x_6x_2^{-1}x_1^{-1}) \\
 &\xrightarrow{(N1)_1} (x_1x_2, x_2x_3, x_3x_2x_3, x_4x_5, x_5x_2x_3, x_6x_2^{-1}x_1^{-1})
 \end{aligned}$$

Hence, it is

$$\begin{aligned}
 f_{x_3} : F &\rightarrow F \\
 x_1 &\mapsto x_1x_2, \\
 x_2 &\mapsto x_2x_3, \\
 x_3 &\mapsto x_3x_2x_3, \\
 x_4 &\mapsto x_4x_5, \\
 x_5 &\mapsto x_5x_2x_3, \\
 x_6 &\mapsto x_6x_2^{-1}x_1^{-1}.
 \end{aligned}$$

- Automorphism  $f_{x_4}$ :

$$\begin{aligned}
 (x_1, x_2, x_3, x_4, x_5, x_6) &\xrightarrow{[(N2)_{1,2}]^2} (x_1x_2^2, x_2, x_3, x_4, x_5, x_6) \\
 &\xrightarrow{(N2)_{2,5}} (x_1x_2^2, x_2x_5, x_3, x_4, x_5, x_6) \\
 &\xrightarrow{(N1)_3} (x_1x_2^2, x_2x_5, x_3^{-1}, x_4, x_5, x_6) \\
 &\xrightarrow{(N2)_{3,2}} (x_1x_2^2, x_2x_5, x_3^{-1}x_2x_5, x_4, x_5, x_6) \\
 &\xrightarrow{(N1)_4} (x_1x_2^2, x_2x_5, x_3^{-1}x_2x_5, x_4^{-1}, x_5, x_6) \\
 &\xrightarrow{(N2)_{5,4}} (x_1x_2^2, x_2x_5, x_3^{-1}x_2x_5, x_4^{-1}, x_5x_4^{-1}, x_6) \\
 &\xrightarrow{(N1)_6} (x_1x_2^2, x_2x_5, x_3^{-1}x_2x_5, x_4^{-1}, x_5x_4^{-1}, x_6^{-1}) \\
 &\xrightarrow{(N2)_{4,6}} (x_1x_2^2, x_2x_5, x_3^{-1}x_2x_5, x_4^{-1}x_6^{-1}, x_5x_4^{-1}, x_6^{-1}) \\
 &\xrightarrow{(N1)_4} (x_1x_2^2, x_2x_5, x_3^{-1}x_2x_5, x_4x_6x_4, x_5x_4^{-1}, x_6^{-1}) \\
 &\xrightarrow{(N2)_{4,5}} (x_1x_2^2, x_2x_5, x_3^{-1}x_2x_5, x_4x_6x_4x_5x_4^{-1}, x_5x_4^{-1}, x_6^{-1}) \\
 &\xrightarrow{(N1)_1} (x_2^{-2}x_1^{-1}, x_2x_5, x_3^{-1}x_2x_5, x_4x_6x_4x_5x_4^{-1}, x_5x_4^{-1}, x_6^{-1}) \\
 &\xrightarrow{(N2)_{6,1}} (x_2^{-2}x_1^{-1}, x_2x_5, x_3^{-1}x_2x_5, x_4x_6x_4x_5x_4^{-1}, x_5x_4^{-1}, x_6^{-1}x_2^{-2}x_1^{-1}) \\
 &\xrightarrow{(N1)_6} (x_2^{-2}x_1^{-1}, x_2x_5, x_3^{-1}x_2x_5, x_4x_6x_4x_5x_4^{-1}, x_5x_4^{-1}, x_1x_2^2x_6)
 \end{aligned}$$

Hence, it is

$$\begin{aligned}
 f_{x_4} : F &\rightarrow F \\
 x_1 &\mapsto x_2^{-2}x_1^{-1}, \\
 x_2 &\mapsto x_2x_5, \\
 x_3 &\mapsto x_3^{-1}x_2x_5, \\
 x_4 &\mapsto x_4x_6x_4x_5x_4^{-1}, \\
 x_5 &\mapsto x_5x_4^{-1}, \\
 x_6 &\mapsto x_1x_2^2x_6.
 \end{aligned}$$

In GAP they define for the automorphisms:

```
#f_{x_1}
fx11:=x1*x2;;
fx12:=x2*x6;;
fx13:=x1*x2*x3;;
fx14:=x3*x4;;
fx15:=x4*x5*x6;;
fx16:=x6*x1*x2;;

#f_{x_2}
fx21:=x1*x3;;
fx22:=x2*x1*x3;;
fx23:=x1*x3^2;;
fx24:=x4*x2;;
fx25:=x5*x6*x2;;
fx26:=x6*x2;;

#f_{x_3}
fx31:=x1*x2;;
fx32:=x2*x3;;
fx33:=x3*x2*x3;;
fx34:=x4*x5;;
fx35:=x5*x2*x3;;
fx36:=x6*x2^-1*x1^-1;;

#f_{x_4}
fx41:=x2^-1*x1^-1;;
fx42:=x2*x5;;
fx43:=x3^-1*x2*x5;;
fx44:=x4*x6*x4*x5*x4^-1;;
fx45:=x5*x4^-1;;
fx46:=x1*x2^2*x6;;
```

Because of the one-to-one correspondence  $a_i \mapsto u_i$ , the ciphertext to the plaintext,

$$S = a_4 a_2 a_1 a_2$$

is here

$$\begin{aligned} C &= c_1 c_2 c_3 c_4 \\ &= f_{x_1}(u_4) f_{x_2}(u_2) f_{x_3}(u_1) f_{x_4}(u_2) \\ &= f_{x_1}((x_4^{-1} x_1^{-1} x_3 x_5)^4) f_{x_2}(x_1^{-1} x_3 x_5 x_6^{-1}) f_{x_3}(x_3^2 x_2^{-1}) f_{x_4}(x_1^{-1} x_3 x_5 x_6^{-1}). \end{aligned}$$

To get the ciphertext Alice calculates in GAP:

```
c1:=(fx14^-1*fx11^-1*fx13*fx15)^4;;
c2:=fx21^-1*fx23*fx25*fx26^-1;;
c3:=fx33^2*fx32^-1;;
c4:=fx41^-1*fx43*fx45*fx46^-1;;

gap> c1;
```

```

(x5*x6)^4
gap> c2;
x3*x5
gap> c3;
x3*x2*x3^2
gap> c4;
x1*x2*x3^-1*x2*x5^2*x4^-1*x6^-1*x2^-2*x1^-1

```

Therefore, it is

$$\begin{aligned}
 C &= c_1 c_2 c_3 c_4 \\
 &= (x_5 x_6)^4 \wr x_3 x_5 \wr x_3 x_2 x_3^2 \wr x_1 x_2 x_3^{-1} x_2 x_5^2 x_4^{-1} x_6^{-1} x_2^{-2} x_1^{-1}.
 \end{aligned}$$

Eve gets from the ciphertext  $C$  the information

$$|c_1| = 8, \quad |c_2| = 2, \quad |c_3| = 4 \quad \text{and} \quad |c_4| = 11.$$

It is

$$L = \sum_{i=1}^4 |c_i| = 25 \quad \text{and} \quad L_1 = \max\{|u_j| \mid 1 \leq j \leq 4\} = 11$$

but the element in  $U$  with maximum length is the element  $u_4 = (x_4^{-1} x_1^{-1} x_3 x_4)^4$ , therefore it is  $\max\{|u_k| \mid u_k \in U\} = |u_4| = 16$ . Hence, it is  $L_1 < \max\{|u_k| \mid u_k \in U\} < L$ . In general, it is more likely that the ball  $B(F, L)$  in the Cayley graph for  $F$  contains a basis for  $F_U$  instead of the ball  $B(F, L_1)$ . Hence, it is likely, that Eve studies the ball  $B(F, L)$  as in Security 7.0.8 assumed.

## B.2. A part of an example with additional information from Alice

Let  $F = \langle X \mid \ \rangle$  be a free group of rank 9 with free generating set  $X = \{x_1, x_2, \dots, x_9\}$ .

In GAP it is:

```
LoadPackage("FGA");;
F:=FreeGroup("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9");;
AssignGeneratorVariables(F);;
```

Let  $f : F \rightarrow F$  be an automorphism of  $F$  which is describable via the following Nielsen transformations

$$(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N2)_{5.6}} (x_1, x_2, x_3, x_4, x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N1)_{5(N1)4}} (x_1, x_2, x_3, x_4^{-1}, x_6^{-1}x_5^{-1}, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N2)_{5.4}} (x_1, x_2, x_3, x_4^{-1}, x_6^{-1}x_5^{-1}x_4^{-1}, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N1)_{5(N1)3}} (x_1, x_2, x_3^{-1}, x_4^{-1}, x_4x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N2)_{4.3}} (x_1, x_2, x_3^{-1}, x_4^{-1}x_3^{-1}, x_4x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N1)_{4}} (x_1, x_2, x_3^{-1}, x_3x_4, x_4x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N2)_{1.2}} (x_1x_2, x_2, x_3^{-1}, x_3x_4, x_4x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N1)_{1}} (x_2^{-1}x_1^{-1}, x_2, x_3^{-1}, x_3x_4, x_4x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N2)_{3.1}} (x_2^{-1}x_1^{-1}, x_2, x_3^{-1}x_2^{-1}x_1^{-1}, x_3x_4, x_4x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N1)_{3(N1)1}} (x_1x_2, x_2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6, x_7, x_8, x_9)$$

$$\xrightarrow{(N1)_{5(N1)7}} (x_1x_2, x_2, x_1x_2x_3, x_3x_4, x_6^{-1}x_5^{-1}x_4^{-1}, x_6, x_7^{-1}, x_8, x_9)$$

$$\xrightarrow{(N2)_{7.5}} (x_1x_2, x_2, x_1x_2x_3, x_3x_4, x_6^{-1}x_5^{-1}x_4^{-1}, x_6, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_8, x_9)$$

$$\xrightarrow{(N1)_{5(N1)8}} (x_1x_2, x_2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_8^{-1}, x_9)$$

$$\xrightarrow{(N2)_{8.7}} (x_1x_2, x_2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_8^{-1}x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_9)$$

$$\xrightarrow{(N1)_{8(N1)9}} (x_1x_2, x_2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_4x_5x_6x_7x_8, x_9^{-1})$$

$$[(N2)_{2.1}]^2 (x_1x_2, x_2(x_1x_2)^2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_4x_5x_6x_7x_8, x_9^{-1})$$

$$\xrightarrow{(N2)_{6.3}} (x_1x_2, x_2(x_1x_2)^2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6x_1x_2x_3, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_4x_5x_6x_7x_8, x_9^{-1})$$

$$\xrightarrow{(N2)_{9.7}} (x_1x_2, x_2(x_1x_2)^2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6x_1x_2x_3, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_4x_5x_6x_7x_8, x_9^{-1}x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1})$$

$$\xrightarrow{(N1)_{9}} (x_1x_2, x_2(x_1x_2)^2, x_1x_2x_3, x_3x_4, x_4x_5x_6, x_6x_1x_2x_3, x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, x_4x_5x_6x_7x_8, x_4x_5x_6x_7x_9);$$

hence the automorphism is

$$\begin{aligned}
 f : F &\rightarrow F \\
 x_1 &\mapsto x_1x_2, \\
 x_2 &\mapsto x_2(x_1x_2)^2, \\
 x_3 &\mapsto x_1x_2x_3, \\
 x_4 &\mapsto x_3x_4, \\
 x_5 &\mapsto x_4x_5x_6, \\
 x_6 &\mapsto x_6x_1x_2x_3, \\
 x_7 &\mapsto x_7^{-1}x_6^{-1}x_5^{-1}x_4^{-1}, \\
 x_8 &\mapsto x_4x_5x_6x_7x_8, \\
 x_9 &\mapsto x_4x_5x_6x_7x_9.
 \end{aligned}$$

Thus, in GAP Alice and Bob define for the automorphism  $f$ :

```

x11:=x1*x2;;
x12:=x2*(x1*x2)^2;;
x13:=x1*x2*x3;;
x14:=x3*x4;;
x15:=x4*x5*x6;;
x16:=x6*x1*x2*x3;;
x17:=x7^-1*x6^-1*x5^-1*x4^-1;;
x18:=x4*x5*x6*x7*x8;;
x19:=x4*x5*x6*x7*x9;;

```

Assume we are in the situation of a cryptosystem as explained in Section 7.1. The alphabet  $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$  consists of 7 elements, hence let  $U$  be a basis for a subgroup of  $F$  with rank 7 and  $U$  is Nielsen reduced.

It is  $U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$  with  $u_1 = x_4x_1^{-1}$ ,  $u_2 = x_1x_4$ ,  $u_3 = x_3^2x_4x_2^{-1}$ ,  $u_4 = x_3x_5$ ,  $u_5 = x_3x_6$ ,  $u_6 = x_3x_8$ ,  $u_7 = x_3x_9$ .

With the operation

▷ `FreeGeneratorsOfGroup(FU)`

which returns a list of free Nielsen reduced generators, which defines a Nielsen reduced set of the finitely generated subgroup  $FU$  of a free group, we can prove that

$$U = \{x_4x_1^{-1}, x_1x_4, x_3^2x_4x_2^{-1}, x_3x_5, x_3x_6, x_3x_8, x_3x_9\}$$

is Nielsen reduced:

```

FU:=Group(x4*x1^-1, x1*x4, x3^2*x4*x2^-1, x3*x5, x3*x6, x3*x8, x3*x9);;

gap> FreeGeneratorsOfGroup(FU);
[ x4*x1^-1, x1*x4, x3^2*x4*x2^-1, x3*x5, x3*x6, x3*x8, x3*x9 ]

```

We assume that Bob encrypted  $c_1, c_2, \dots, c_{i-1}$  correctly and no letters of  $c_i$  are canceled in  $C_{red}^{(i)}$ , it is  $C_{red}^{(i)} = c_{i-1}^{-1} \dots c_2^{-1} c_1^{-1} C_{red}$  with  $C_{red}$  the send ciphertext from Alice. Hence, we assume,

that he gets

$$\begin{aligned} C_{red}^{(i)} &\equiv c_i \omega \\ &\equiv x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 \omega', \end{aligned}$$

with  $\omega$  the reduced word of  $c_{i+1} \cdots c_z$  and  $\omega' = \omega$  or  $\omega'$  is a terminal segment of  $\omega$ . To encrypt  $c_i$  he calculates  $U_{f_{x_i}} = U_f$ . In GAP this is

```
fu1:=x14*x11^-1;;
fu2:=x11*x14;;
fu3:=x13^2*x14*x12^-1;;
fu4:=x13*x15;;
fu5:=x13*x16;;
fu6:=x13*x18;;
fu7:=x13*x19;;

gap> fu1;
x3*x4*x2^-1*x1^-1
gap> fu2;
x1*x2*x3*x4
gap> fu3;
(x1*x2*x3)^2*x3*x4*(x2^-1*x1^-1)^2*x2^-1
gap> fu4;
x1*x2*x3*x4*x5*x6
gap> fu5;
x1*x2*x3*x6*x1*x2*x3
gap> fu6;
x1*x2*x3*x4*x5*x6*x7*x8
gap> fu7;
x1*x2*x3*x4*x5*x6*x7*x9
```

and therefore it is

$$\begin{aligned} U_{f_{x_i}} = U_f &= \{f(x_4 x_1^{-1}), f(x_1 x_4), f(x_3^2 x_4 x_2^{-1}), f(x_3 x_5), f(x_3 x_6), f(x_3 x_8), f(x_3 x_9)\} \\ &= \{x_3 x_4 x_2^{-1} x_1^{-1}, x_1 x_2 x_3 x_4, (x_1 x_2 x_3)^2 x_3 x_4 (x_2^{-1} x_1^{-1})^2 x_2^{-1}, \\ &\quad x_1 x_2 x_3 x_4 x_5 x_6, x_1 x_2 x_3 x_6 x_1 x_2 x_3, x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8, x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_9\} \end{aligned}$$

and he knows

$$\begin{aligned} u_1 &\mapsto x_3 x_4 x_2^{-1} x_1^{-1}, \\ u_2 &\mapsto x_1 x_2 x_3 x_4, \\ u_3 &\mapsto (x_1 x_2 x_3)^2 x_3 x_4 (x_2^{-1} x_1^{-1})^2 x_2^{-1}, \\ u_4 &\mapsto x_1 x_2 x_3 x_4 x_5 x_6, \\ u_5 &\mapsto x_1 x_2 x_3 x_6 x_1 x_2 x_3, \\ u_6 &\mapsto x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8, \\ u_7 &\mapsto x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_9. \end{aligned}$$

If he now looks at  $C_{red}^{(i)}$  he is not able to decide if the ciphertext unit  $c_i$  encrypt  $a_2$  or  $a_4$  or  $a_6$ .

Hence, Alice has to send additional information to Bob.

If Bob gets the information

- $(i, 4, 1)$  he knows, that the element  $c_i$  is of free length 4. There are two elements in  $U_f$  of free length 4, which are  $f(u_1)$  and  $f(u_2)$ , but Bob knows that all 4 elements are visible as initial segment of  $C_{red}^{(i)}$  and hence  $c_i = x_1x_2x_3x_4 = f(u_2)$ , thus  $s_i = a_2$ ;
- $(i, 6, 1)$  he knows, that the element  $c_i$  is of free length 6 and hence he knows that now  $c_i = x_1x_2x_3x_4x_5x_6 = f(u_4)$ , thus  $s_i = a_4$ ;
- $(i, 7, x_8)$  he knows, that the element  $c_i$  is of free length at least 8 and the 8th letter differs from all other elements in  $U_f$ , which have the same first seven letters as  $c_i$  and these seven letters are an initial segment of  $C_{red}^{(i)}$ , and the 8th letter is  $x_8$ . Hence, he knows, that  $c_i = x_1x_2x_3x_4x_5x_6x_7x_8 = f(u_6)$ , thus  $s_i = a_6$ ;
- $(i, 8, 1)$  he knows, that  $|c_i| = 8$  and that all 8 letters of  $c_i$  are the first letters of  $C_{red}^{(i)}$  hence he knows, that  $c_i = x_1x_2x_3x_4x_5x_6x_7x_8 = f(u_6)$ , thus  $s_i = a_6$ .

**Example B.2.1.** Assume the ciphertext unit  $c_i$  is a reduced initial segment of the word  $C_{red}^{(i)}$ , that means, there are cancellations between  $c_i$  and  $c_{i+1}$ . Hence, let

$$C_{red}^{(i)} \equiv x_1x_2x_3x'\omega,$$

with  $x' \in X^{\pm 1} \setminus \{x_3^{-1}, x_4, x_1\}$  and  $\omega$ , be a word in  $X$ . If Bob gets the information

- $(i, 3, x_41)$ ,  $x_41$  means that after  $x_4$  is the empty word, and hence it is known that the free length of  $c_i$  is  $|c_i| = 3 + |x_41| = 4$  and thus  $c_i$  has the same first 3 letters as  $C_{red}^{(i)}$  and the terminal segment of  $c_i$  is  $x_4$ . Therefore, Bob knows that  $c_i = f(u_2)$ , thus  $s_i = a_2$ ;
- $(i, 13, 1)$  ( $u_3$  is the only element in  $U_f$  of free length 13) or  $(i, 3, x_1)$  he knows  $|c_i| > 3 + 1$ , the first 3 letters are the same first 3 letters of  $C_{red}^{(i)}$  and the 4th letter differs from every other 4th letter of each element in  $U_f$ , which first 3 letters are the same as the first 3 letters of  $C_{red}^{(i)}$ . Therefore, Bob knows  $c_i = f(u_3)$ , thus  $s_i = a_3$ .
- $(i, 6, 1)$  ( $u_4$  is the only element in  $U_f$  of free length 6) or  $(i, 3, x_4x_5x_61)$ ,  $x_4x_5x_61$  means that after  $x_4x_5x_6$  is the empty word, and hence it is known that  $|c_i| = 3 + |x_4x_5x_61| = 6$  and thus  $c_i$  has the same first 3 letters as  $C_{red}^{(i)}$  and the terminal segment of  $c_i$  is  $x_4x_5x_6$ . Therefore, he knows that  $c_i = f(u_4)$ , thus  $s_i = a_4$ ;
- $(i, 8, 1)$  he does not know if  $c_i = f(u_6)$  or  $c_i = f(u_7)$  hence Alice has to give one of the following information to Bob
  1.  $(i, 3, x_4x_5x_6x_7x_81)$  if  $c_i = f(u_6)$  and thus  $s_i = a_6$ . The word  $x_4x_5x_6x_7x_81$  means that after  $x_4x_5x_6x_7x_8$  is the empty word, and hence it is known that for the free length of  $c_i$  it is  $|c_i| = 3 + |x_4x_5x_6x_7x_81| = 8$  and thus  $c_i$  has the same first 3 letters as  $C_{red}^{(i)}$  and the terminal segment of  $c_i$  is  $x_4x_5x_6x_7x_8$ .
  2.  $(i, 3, x_4x_5x_6x_7x_91)$  if  $c_i = f(u_7)$  and thus  $s_i = a_7$ . The word  $x_4x_5x_6x_7x_91$  means that after  $x_4x_5x_6x_7x_9$  is the empty word, and hence it is known that for the free length of  $c_i$  it is  $|c_i| = 3 + |x_4x_5x_6x_7x_91| = 8$  and thus  $c_i$  has the same first 3 letters as  $C_{red}^{(i)}$  and the terminal segment of  $c_i$  is  $x_4x_5x_6x_7x_9$ .



### B.3. Example for Remark 7.0.9

Let  $F$  be a finitely generated free group on the free generating set  $X = \{a, b, c, d\}$ . We give a small example for a Nielsen reduced set  $U = \{u_1, u_2\}$ , with  $u_1, u_2$  words in  $X$ , and two automorphisms  $f_1, f_2 \in \text{Aut}(F)$ , such that for  $c_1 = f_1(u_1)$  and  $c_2 = f_2(u_2)$  the following holds:

$$L = \sum_{i=1}^2 |c_i| < \max\{|u_1|, |u_2|\}$$

In this case not all elements of  $U$  can be found in a ball  $B(F, L)$  in the Cayley graph of  $F$ .

Let  $u_1 = (ab)^3$  and  $u_2 = cd$  then  $U = \{(ab)^3, cd\}$  is a Nielsen reduced set (Lemma 4.2.15 is fulfilled).

Let  $f_1 : F \rightarrow F$  be an automorphism of  $F$ , which is describable via the following Nielsen transformations

$$\begin{aligned} & (a, b, c, d) \\ & \xrightarrow{(N1)_1(N1)_3} (a^{-1}, b, c^{-1}, d) \\ & \xrightarrow{(N2)_{1,3}} (a^{-1}c^{-1}, b, c^{-1}, d) \\ & \xrightarrow{(N1)_1} (ca, b, c^{-1}, d) \\ & \xrightarrow{(N2)_{3,1}} (ca, b, a, d) \\ & \xrightarrow{(N1)_3} (ca, b, a^{-1}, d) \\ & \xrightarrow{(N2)_{1,4}} (c, b, a^{-1}, d) \\ & \xrightarrow{(N1)_2} (c, b^{-1}, a^{-1}, d) \\ & \xrightarrow{(N2)_{2,1}} (c, b^{-1}c, a^{-1}, d) \\ & \xrightarrow{(N2)_{2,4}} (c, b^{-1}cd, a^{-1}, d) \\ & \xrightarrow{(N1)_2} (c, d^{-1}c^{-1}b, a^{-1}, d) \\ & \xrightarrow{(N2)_{1,4}} (cd, d^{-1}c^{-1}b, a^{-1}, d) \\ & \xrightarrow{[(N2)_{3,4}]^3} (cd, d^{-1}c^{-1}b, a^{-1}d^3, d) \\ & \xrightarrow{(N2)_{4,1}} (cd, d^{-1}c^{-1}b, a^{-1}d^3, dcd); \end{aligned}$$

hence the automorphism is

$$\begin{aligned} f_1 : F &\rightarrow F \\ a &\mapsto cd, \\ b &\mapsto d^{-1}c^{-1}b, \\ c &\mapsto a^{-1}d^3, \\ d &\mapsto dcd. \end{aligned}$$

Let  $f_2 : F \rightarrow F$  be an automorphism of  $F$ , which is describable via the following Nielsen

transformations

$$\begin{aligned}
 & (a, b, c, d) \\
 & \xrightarrow{(N1)_1(N1)_3} (a^{-1}, b, c^{-1}, d) \\
 & \xrightarrow{(N2)_{3,4}} (a^{-1}, b, c^{-1}d, d) \\
 & \xrightarrow{(N1)_4} (a^{-1}, b, c^{-1}d, d^{-1}) \\
 & \xrightarrow{(N2)_{4,1}} (a^{-1}, b, c^{-1}d, d^{-1}a^{-1}) \\
 & \xrightarrow{(N2)_{2,4}} (a^{-1}, bd^{-1}a^{-1}, c^{-1}d, d^{-1}a^{-1}) \\
 & \xrightarrow{(N1)_1} (a, bd^{-1}a^{-1}, c^{-1}d, d^{-1}a^{-1}) \\
 & \xrightarrow{(N2)_{1,3}} (ac^{-1}d, bd^{-1}a^{-1}, c^{-1}d, d^{-1}a^{-1});
 \end{aligned}$$

hence the automorphism is

$$\begin{aligned}
 f_2 : F &\rightarrow F \\
 a &\mapsto ac^{-1}d, \\
 b &\mapsto bd^{-1}a^{-1}, \\
 c &\mapsto c^{-1}d, \\
 d &\mapsto d^{-1}a^{-1}.
 \end{aligned}$$

It is

$$c_1 = f_1(u_1) = f_1((ab)^3) = (cdd^{-1}c^{-1}b)^3 = b^3$$

and

$$c_2 = f_2(u_2) = f_2(cd) = c^{-1}dd^{-1}a^{-1} = c^{-1}a^{-1}.$$

Now,  $L = |c_1| + |c_2| = 3 + 2 = 5$  and  $\max\{|u_1|, |u_2|\} = \max\{6, 2\} = 6$ , thus

$$5 = L < \max\{|u_1|, |u_2|\} = 6.$$

## Appendix C

### Calculations with Maple 16 or GAP for examples

If there are Nielsen transformations of type  $(N1)$  one after another we can apply them in one step. For example if the Nielsen transformations  $(N1)_5$   $(N1)_2$   $(N1)_1$   $(N2)_{3.2}$  are applied to a set  $(a, b, c, d, e)$  we write instead of

$$\begin{aligned} (a, b, c, d, e) &\xrightarrow{(N1)_5} (a, b, c, d, e^{-1}) \\ &\xrightarrow{(N1)_2} (a, b^{-1}, c, d, e^{-1}) \\ &\xrightarrow{(N1)_1} (a^{-1}, b^{-1}, c, d, e^{-1}) \\ &\xrightarrow{(N2)_{3.2}} (a^{-1}, b^{-1}, cb^{-1}, d, e^{-1}) \end{aligned}$$

the following

$$\begin{aligned} (a, b, c, d, e) &\xrightarrow{(N1)_5(N1)_2(N1)_1} (a^{-1}, b^{-1}, c, d, e^{-1}) \\ &\xrightarrow{(N2)_{3.2}} (a^{-1}, b^{-1}, cb^{-1}, d, e^{-1}). \end{aligned}$$

We use the FGA<sup>1</sup> package in GAP<sup>2</sup>.

In Maple 16<sup>3</sup> the package “LinearAlgebra” is used.

#### C.1. Example 2.1.5 calculations in Maple 16

First, Alice and Bob have to agree on a private key.

The real inner product space is  $W = \mathbb{R}^6$  and the subspace  $V$  is of dimension 3. In Maple 16 they define:

```
> restart;
> with(LinearAlgebra):
> m:=6: t:=3:
> B:=RandomMatrix(t,m);
```

$$B := \begin{bmatrix} 66 & 20 & -34 & -21 & -50 & -79 \\ -36 & -7 & -62 & -56 & 30 & -71 \\ -41 & 16 & -90 & -8 & 62 & 28 \end{bmatrix}$$

```
> Rank(B);
```

3

<sup>1</sup>Free Group Algorithms. A GAP4 Package by Christian Sievers, TU Braunschweig.

<sup>2</sup>Groups, Algorithms and Programming [GAP15]

<sup>3</sup>More precisely Classic Worksheet Maple 16 is used.

The rows of the matrix  $B$  define a basis for a 3-dimensional subspace  $V$  in  $\mathbb{R}^6$ .

Bob calculates for encryption his private key, which is a basis for the orthogonal complement  $V^\perp$  to  $V$ . This is the nullspace of the matrix  $B$ :

```
> kern:=NullSpace(B);
```

$$kern := \left\{ \begin{bmatrix} \frac{67528}{61217} \\ -\frac{72236}{61217} \\ -\frac{1433}{61217} \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{129349}{61217} \\ -\frac{362352}{61217} \\ -\frac{208597}{122434} \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{46287}{61217} \\ -\frac{191417}{61217} \\ -\frac{121115}{122434} \\ 1 \\ 0 \\ 0 \end{bmatrix} \right\}$$

Alice needs as decryption key an orthonormal basis of the subspace, which she gets with the help of the Gram-Schmidt procedure and a normalization:

```
> L:= [seq(B[j,1..m], j=1..t)];
> G:=GramSchmidt(L,normalized);
```

$$G := \left[ \begin{bmatrix} \frac{33\sqrt{15094}}{7547}, \frac{10\sqrt{15094}}{7547}, -\frac{17\sqrt{15094}}{7547}, -\frac{21\sqrt{15094}}{15094}, -\frac{25\sqrt{15094}}{7547}, -\frac{79\sqrt{15094}}{15094} \\ -\frac{144211\sqrt{12849748610}}{32124371525}, -\frac{101599\sqrt{12849748610}}{96373114575}, -\frac{25667\sqrt{12849748610}}{6424874305} \\ -\frac{742847\sqrt{12849748610}}{192746229150}, \frac{69667\sqrt{12849748610}}{19274622915}, -\frac{228797\sqrt{12849748610}}{64248743050} \\ \frac{1141824503\sqrt{1211483474194866265}}{6057417370974331325}, \frac{5968746002\sqrt{1211483474194866265}}{18172252112922993975} \\ -\frac{830820734\sqrt{1211483474194866265}}{1211483474194866265}, \frac{3745334828\sqrt{1211483474194866265}}{18172252112922993975} \\ \frac{594555334\sqrt{1211483474194866265}}{3634450422584598795}, \frac{2286429878\sqrt{1211483474194866265}}{6057417370974331325} \end{bmatrix}, \left[ \right]$$

Bob wants to send Alice a message  $p \in \mathbb{R}^6$ . He also chooses an element  $w \in V$  and calculates  $v = w - p$ :

```
> p:=Transpose(<3,18,25,16,20,15>);
> a:=3: b:=-6: c:=7:
> w:=a*B[1,1..m]+b*B[2,1..m]+c*B[3,1..m];
> v:=w-p;
```

$$p := [3, 18, 25, 16, 20, 15]$$

$$w := [127, 214, -360, 217, 104, 385]$$

$$v := [124, 196, -385, 201, 84, 370]$$

In addition he needs the element  $w^*$  with the property, that  $w$  is the closest vector in  $V$  to  $w^*$ . Therefore, he needs his encryption key stored as  $kern$ , which is a basis for the orthogonal complement  $V^\perp$ . He stores  $w^*$  as  $ww$ , and sends  $(ww, v)$  to Alice:

```

> w:=Transpose(w):
> r:=m-t:
> R:=RandomVector(r):
>   while Equal(R,Vector(r)) do
>     R:=RandomVector(r):
>   end:
> R;

```

$$\begin{bmatrix} 13 \\ -65 \\ 5 \end{bmatrix}$$

```

> u:=Vector(m):
> for k from 1 to r do
>   u:= u + kern[k]*R[k]:
> end:
> ww:=u+w;

```

$$ww := \begin{bmatrix} \frac{476173}{61217} \\ \frac{34757165}{61217} \\ -15580134 \\ \frac{61217}{61217} \\ 222 \\ 117 \\ 320 \end{bmatrix}$$

Alice gets  $(ww, v)$ . With her decryption key (an orthonormal basis for  $V$ ), which is stored in  $G$ , she is able to calculate  $U = w$  with the Closest Vector Theorem. Afterwards she calculates  $P = U - v$ , which is the message from Bob:

```

> V:=Transpose(Vector(m)):
> for k from 1 to t do
>   U:=U+DotProduct(ww,G[k])*G[k]:
> end:
> P:=U-v;

```

$$P := [3, 18, 25, 16, 20, 15]$$

## C.2. Example 2.2.2 calculations in Maple 16

In a challenge and response system, the prover and the verifier agree privately on a password and a challenge subspace  $V$ . In this example they agree on the password  $P = Alice$  and the challenge space  $V$  is a 3-dimensional subspace of  $\mathbb{R}^7$ . In Maple 16 they define:

```
> restart;
> with(LinearAlgebra):
> m:=7: t:=3:
> B:=RandomMatrix(t,m);
      B := 
$$\begin{bmatrix} 13 & 66 & 20 & -34 & -21 & -50 & -79 \\ -65 & -36 & -7 & -62 & -56 & 30 & -71 \\ 5 & -41 & 16 & -90 & -8 & 62 & 28 \end{bmatrix}$$

> Rank(B);
```

3

The rows of the matrix  $B$  define a basis for the 3-dimensional subspace of  $\mathbb{R}^7$ , which is the common subspace  $V$ .

The prover should verify himself to the verifier. For this, he sends the password  $P = Alice$  to the verifier. Now, the verifier knows the corresponding challenge space  $V$  to the password  $P$  and generates a challenge for the prover. He chooses three elements  $w, u, p \in V$  and asks for the length of the “line” between these elements. For  $w, u, p \in V$  he chooses:

```
> a1:=2: a2:=4: a3:=-2:
> b1:=5: b2:=-3: b3:=2:
> c1:=3: c2:=-2: c3:=-1:
> w:=a1*B[1,1..m]+a2*B[2,1..m]+a3*B[3,1..m];
> u:=b1*B[1,1..m]+b2*B[2,1..m]+b3*B[3,1..m];
> p:=c1*B[1,1..m]+c2*B[2,1..m]+c3*B[3,1..m];
      w := [-244, 70, -20, -136, -250, -104, -498]
      u := [270, 356, 153, -164, 47, -216, -126]
      p := [164, 311, 58, 112, 57, -272, -123]
```

Now, he needs the challenge elements for the prover. He first calculates a basis for the orthogonal complement  $V^\perp$  of  $V$  and stored it as *kern*:

```
> kern:=NullSpace(B);
      kern := 
$$\left\{ \begin{bmatrix} \frac{-208597}{112011} \\ \frac{40305}{37337} \\ \frac{179012}{112011} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{-121115}{112011} \\ \frac{5827}{37337} \\ \frac{138649}{112011} \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{-2866}{112011} \\ \frac{40656}{37337} \\ \frac{-120604}{112011} \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{-122434}{112011} \\ \frac{-22648}{37337} \\ \frac{494216}{112011} \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

```

He calculates the elements  $w^* =: XX$ ,  $u^* =: YY$  and  $p^* =: ZZ$  for which  $w$ ,  $u$ , and  $p$ , respectively, are the closest elements in  $V$ , as follows:

```

> r:=m-t:
> R:=RandomVector(r):
> S:=RandomVector(r):
> E:=RandomVector(r):
> while Equal(R,Vector(r)) do
>     R:=RandomVector(r):
> end:
> while Equal(S,Vector(r)) do
>     S:=RandomVector(r):
> end:
> while Equal(E,Vector(r)) do
>     E:=RandomVector(r):
> end:
> R;
> S;
> E;

```

$$\begin{bmatrix} -66 \\ 55 \\ 68 \\ 26 \end{bmatrix}$$

$$\begin{bmatrix} -75 \\ 38 \\ 97 \\ -82 \end{bmatrix}$$

$$\begin{bmatrix} -69 \\ 23 \\ 25 \\ 5 \end{bmatrix}$$

```

> x:=Vector(m):
> for k from 1 to r do
>     x:= x + kern[k]*R[k]:
> end:
> XX:=Transpose(w)+x;
> y:=Vector(m):
> for k from 1 to r do
>     y:= y + kern[k]*S[k]:
> end:
> YY:=y+Transpose(u);
> z:=Vector(m):
> for k from 1 to r do
>     z:= z + kern[k]*E[k]:
> end:
> ZZ:=Transpose(p)+z;

```

$$XX := \begin{bmatrix} -7867593 \\ 37337 \\ 2449705 \\ 37337 \\ -593591 \\ 37337 \\ -110 \\ -195 \\ -36 \\ -564 \end{bmatrix}$$

$$\begin{aligned}
 YY &:= \begin{bmatrix} \frac{51046961}{112011} \\ \frac{16291291}{37337} \\ -43243855 \\ 112011 \\ -246 \\ 85 \\ -119 \\ -201 \end{bmatrix} \\
 ZZ &:= \begin{bmatrix} \frac{29293532}{112011} \\ \frac{9867943}{37337} \\ -3210283 \\ 112011 \\ 117 \\ 80 \\ -247 \\ -192 \end{bmatrix}
 \end{aligned}$$

It is  $XX = w^* = w + (-66u_1^\perp + 55u_2^\perp + 68u_3^\perp + 26u_4^\perp)$ ,  
 $YY = u^* = u + (-75u_1^\perp + 38u_2^\perp + 97u_3^\perp - 82u_4^\perp)$  and  
 $ZZ = p^* = p + (-69u_1^\perp + 23u_2^\perp + 25u_3^\perp + 5u_4^\perp)$ , with  $kern = \{u_1^\perp, u_2^\perp, u_3^\perp, u_4^\perp\}$ .

The Verifier sends the challenge  $(XX, YY, ZZ)$  with a computational accuracy of 12 digits to the prover.

To calculate the response the prover first uses the Gram-Schmidt procedure and a normalization to generate a orthonormal basis  $G$  for the subspace  $V$ :

```

> L:= [seq(B[j, 1..m], j=1..t)]:
> G:=GramSchmidt(L, normalized);

```



$$\begin{aligned}
G := & \left[ \left[ \frac{13\sqrt{15263}}{15263}, \frac{66\sqrt{15263}}{15263}, \frac{20\sqrt{15263}}{15263}, -\frac{34\sqrt{15263}}{15263}, -\frac{21\sqrt{15263}}{15263}, -\frac{50\sqrt{15263}}{15263}, \right. \right. \\
& \left. \left. -\frac{79\sqrt{15263}}{15263} \right], \left[ -\frac{1044511\sqrt{4059517036667}}{4059517036667}, -\frac{815580\sqrt{4059517036667}}{4059517036667}, \right. \right. \\
& \left. \left. -\frac{187481\sqrt{4059517036667}}{4059517036667}, -\frac{809218\sqrt{4059517036667}}{4059517036667}, \right. \right. \\
& \left. \left. -\frac{770056\sqrt{4059517036667}}{4059517036667}, \frac{659490\sqrt{4059517036667}}{4059517036667}, \right. \right. \\
& \left. \left. -\frac{765145\sqrt{4059517036667}}{4059517036667} \right], \left[ \frac{10791066889\sqrt{10738342994104662523}}{85906743952837300184}, \right. \right. \\
& \left. \left. \frac{769777141\sqrt{10738342994104662523}}{85906743952837300184}, \frac{7309852223\sqrt{10738342994104662523}}{85906743952837300184}, \right. \right. \\
& \left. \left. -\frac{9995226829\sqrt{10738342994104662523}}{42953371976418650092}, \right. \right. \\
& \left. \left. \frac{2499753517\sqrt{10738342994104662523}}{85906743952837300184}, \frac{3654268049\sqrt{10738342994104662523}}{42953371976418650092}, \right. \right. \\
& \left. \left. \frac{3791392411\sqrt{10738342994104662523}}{42953371976418650092} \right] \right]
\end{aligned}$$

With the Closest Vector Theorem he is now able to reconstruct the elements  $w$ ,  $u$  and  $p$  from  $XX$ ,  $YY$ ,  $ZZ$ , respectively, which he stores as  $W$ ,  $U$  and  $P$ :

```

> W:=Transpose(Vector(m)):
> for k from 1 to t do
>     W:=W+DotProduct(XX,G[k])*G[k]:
> end:
> W;
[-244, 70, -20, -136, -250, -104, -498]

> U:=Transpose(Vector(m)):
> for k from 1 to t do
>     U:=U+DotProduct(YY,G[k])*G[k]:
> end:
> U;
[270, 356, 153, -164, 47, -216, -126]

> P:=Transpose(Vector(m)):
> for k from 1 to t do
>     P:=P+DotProduct(ZZ,G[k])*G[k]:
> end:
> P;
[164, 311, 58, 112, 57, -272, -123]

```

The prover calculates the “line” between his computed elements in  $V$  with a computational accuracy of 12 digits:

```

> Digits:=12:
> nnn:=evalf(Norm(W-U,2)+Norm(U-P,2)+Norm(P-W,2));
nnn := 1848.80960451

```

He sends  $nnn := 1848.80960451$  as response to the verifier. The verifier compares this with his length of the “line”, which is:

```
> Digits:=12;
> nn:=evalf(Norm(w-u,2)+Norm(u-p,2)+Norm(p-w,2));
      nn := 1848.80960451
```

It is  $nn = nnn$ , therefore the response is correct. It is not possible to generate the correct response by knowing just  $XX$ ,  $YY$  and  $ZZ$ . The “line” between these elements is  $nnnn = 2319.52030261$  which is not  $nn = 1848.80960451$ :

```
> Digits:=12;
> nnnn:=evalf(Norm(XX-YY,2)+Norm(YY-ZZ,2)+Norm(ZZ-XX,2));
      Digits := 12
      nnnn := 2319.52030261
```

### C.3. Example 6.1.3 calculations in Maple 16

We now present the calculations which were needed to execute Example 6.1.3, which is a (3, 2)-secret sharing scheme using free subgroups in  $SL(2, \mathbb{Q})$ .

We take a closer look at the steps for the dealer for this (3, 2)-secret sharing scheme. It is  $n = 3$  and  $t = 2$ . Thus, in Maple we define:

```
> restart; with(LinearAlgebra):
> n := 3; t := 2; m := binomial(n, t-1);
      n := 3
      t := 2
      m := 3
```

The dealer needs  $m = 3$  matrices. For this he chooses rational numbers  $r_i =: r[i]$ ,  $1 \leq i \leq 3$ , with the properties (4.1). Hence, these rational numbers were chosen as follows and the inequalities (4.1) were proved:

```
> r[1] := 7/2; r[2] := 15/2; r[3] := 11;
      r1 := 7/2
      r2 := 15/2
      r3 := 11
> r[1]-2; r[2]-r[1]-3; r[3]-r[2]-3;
      3/2
      1
      1/2
```

All results are greater than 0, hence he can generate with the numbers  $r_1$ ,  $r_2$  and  $r_3$  matrices, which generate a free subgroup of  $SL(2, \mathbb{Q})$  of rank 3. The matrices are:

```
> M[1] := Matrix([[ -r[1], r[1]^2-1], [1, -r[1]]]);
> M[2] := Matrix([[ -r[2], r[2]^2-1], [1, -r[2]]]);
> M[3] := Matrix([[ -r[3], r[3]^2-1], [1, -r[3]]]);
```

$$M_1 := \begin{bmatrix} -\frac{7}{2} & \frac{45}{4} \\ 1 & -\frac{7}{2} \end{bmatrix}$$

$$M_2 := \begin{bmatrix} -\frac{15}{2} & \frac{221}{4} \\ 1 & -\frac{15}{2} \end{bmatrix}$$

$$M_3 := \begin{bmatrix} -11 & 120 \\ 1 & -11 \end{bmatrix}$$

The secret is reconstructible with the help of the traces of these matrices, which are:

```
> Trace(M[1]); Trace(M[2]); Trace(M[3]);
      -7
      -15
      -22
```

To get the Nielsen equivalent set  $N$  and  $U$ , from which the dealer constructs the share-sets for the participants, he does the following elementary Nielsen transformations on the matrices and on an abstract set. These matrix multiplications are the following:

```

> M[11] := M[1]; M[12] := M[2]; M[13] := M[3];

```

$$M_{11} := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$

$$M_{12} := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$M_{13} := \begin{bmatrix} -11 & 120 \\ 1 & -11 \end{bmatrix}$$

```

> M[21] := M[11]; M[22] := MatrixInverse(M[12]); M[23] := M[13];

```

$$M_{21} := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$

$$M_{22} := \begin{bmatrix} \frac{-15}{2} & \frac{-221}{4} \\ -1 & \frac{-15}{2} \end{bmatrix}$$

$$M_{23} := \begin{bmatrix} -11 & 120 \\ 1 & -11 \end{bmatrix}$$

```

> M[31] := M[21].M[22]; M[32] := M[22]; M[33] := M[23];

```

$$M_{31} := \begin{bmatrix} 15 & 109 \\ -4 & -29 \end{bmatrix}$$

$$M_{32} := \begin{bmatrix} \frac{-15}{2} & \frac{-221}{4} \\ -1 & \frac{-15}{2} \end{bmatrix}$$

$$M_{33} := \begin{bmatrix} -11 & 120 \\ 1 & -11 \end{bmatrix}$$

```

> M[41] := M[31]; M[42] := M[32]; M[43] := M[33].M[32]^3;

```

$$M_{41} := \begin{bmatrix} 15 & 109 \\ -4 & -29 \end{bmatrix}$$

$$M_{42} := \begin{bmatrix} \frac{-15}{2} & \frac{-221}{4} \\ -1 & \frac{-15}{2} \end{bmatrix}$$

$$M_{43} := \begin{bmatrix} -8565 & -63664 \\ 799 & 5939 \end{bmatrix}$$

```

> M[51] := M[41]; M[52] := M[42].M[43]; M[53] := M[43];

```

$$M_{51} := \begin{bmatrix} 15 & 109 \\ -4 & -29 \end{bmatrix}$$

$$M_{52} := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$M_{53} := \begin{bmatrix} -8565 & -63664 \\ 799 & 5939 \end{bmatrix}$$

> M[61] := MatrixInverse(M[51]); M[62] := M[52]; M[63] := M[53];

$$M_{61} := \begin{bmatrix} -29 & -109 \\ 4 & 15 \end{bmatrix}$$

$$M_{62} := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$M_{63} := \begin{bmatrix} -8565 & -63664 \\ 799 & 5939 \end{bmatrix}$$

> M[71] := M[61].M[62]; M[72] := M[62]; M[73] := M[63];

$$M_{71} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$M_{72} := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$M_{73} := \begin{bmatrix} -8565 & -63664 \\ 799 & 5939 \end{bmatrix}$$

> M[81] := M[71]; M[82] := M[72]; M[83] := MatrixInverse(M[73]);

$$M_{81} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$M_{82} := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$M_{83} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> M[91] := M[81]; M[92] := M[82]; M[93] := M[83].M[82];

$$M_{91} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$M_{92} := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$M_{93} := \begin{bmatrix} \frac{1132425929}{4} & \frac{8417369243}{4} \\ -\frac{152350279}{4} & -\frac{1132425989}{4} \end{bmatrix}$$

> R[1] := M[91]; R[2] := M[92]; R[3] := M[93];

$$R_1 := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$R_2 := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$R_3 := \begin{bmatrix} \frac{1132425929}{4} & \frac{8417369243}{4} \\ -\frac{152350279}{4} & -\frac{1132425989}{4} \end{bmatrix}$$

The share distribution is done by the method given by D. Panagopoulos.

If now two or more participants combine their shares they get the following matrices:

> R[1]; R[2]; R[3];

$$\begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$\begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1132425929}{4} & \frac{8417369243}{4} \\ -\frac{152350279}{4} & -\frac{1132425989}{4} \end{bmatrix}$$

Next, the calculations for the participants to generate a set  $M'$  are given. They do Nielsen transformations on the explicit set of matrices, which they reconstruct from the theoretical set.

These are the following matrix operations:

> R[11] := R[1]; R[12] := R[2]; R[13] := R[3];

$$R_{11} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$R_{12} := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$R_{13} := \begin{bmatrix} \frac{1132425929}{4} & \frac{8417369243}{4} \\ -\frac{152350279}{4} & -\frac{1132425989}{4} \end{bmatrix}$$

> R[21] := R[11]; R[22] := MatrixInverse(R[12]); R[23] := R[13];

$$R_{21} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$R_{22} := \begin{bmatrix} \frac{38243}{2} & \frac{-597401}{4} \\ \frac{-5145}{2} & \frac{80371}{4} \end{bmatrix}$$

$$R_{23} := \begin{bmatrix} \frac{1132425929}{4} & \frac{8417369243}{4} \\ \frac{-152350279}{4} & \frac{-1132425989}{4} \end{bmatrix}$$

> R[31] := R[21]; R[32] := R[22]; R[33] := R[23].R[22];

$$R_{31} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$R_{32} := \begin{bmatrix} \frac{38243}{2} & \frac{-597401}{4} \\ \frac{-5145}{2} & \frac{80371}{4} \end{bmatrix}$$

$$R_{33} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> R[41] := R[31]; R[42] := MatrixInverse(R[32]); R[43] := R[33];

$$R_{41} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$R_{42} := \begin{bmatrix} \frac{80371}{4} & \frac{597401}{4} \\ \frac{5145}{2} & \frac{38243}{2} \end{bmatrix}$$

$$R_{43} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> R[51] := R[41]; R[52] := R[42].R[43]; R[53] := R[43];

$$R_{51} := \begin{bmatrix} \frac{-3452369}{4} & \frac{-25661603}{4} \\ \frac{237917}{2} & \frac{1768447}{2} \end{bmatrix}$$

$$R_{52} := \begin{bmatrix} \frac{-15}{2} & \frac{-221}{4} \\ -1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{53} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> R[61] := R[51].R[53]; R[62] := R[52]; R[63] := R[53];

$$R_{61} := \begin{bmatrix} \frac{653}{2} & \frac{9679}{4} \\ -45 & \frac{-667}{2} \end{bmatrix}$$

$$R_{62} := \begin{bmatrix} \frac{-15}{2} & \frac{-221}{4} \\ -1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{63} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> R[71] := R[61]; R[72] := MatrixInverse(R[62]); R[73] := R[63];

$$R_{71} := \begin{bmatrix} \frac{653}{2} & \frac{9679}{4} \\ -45 & \frac{-667}{2} \end{bmatrix}$$

$$R_{72} := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{73} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> R[81] := R[71].R[72]; R[82] := R[72]; R[83] := R[73];

$$R_{81} := \begin{bmatrix} -29 & -109 \\ 4 & 15 \end{bmatrix}$$

$$R_{82} := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{83} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> R[91] := MatrixInverse(R[81]); R[92] := R[82]; R[93] := R[83];

$$R_{91} := \begin{bmatrix} 15 & 109 \\ -4 & -29 \end{bmatrix}$$

$$R_{92} := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{93} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

> R[101] := R[91].R[92]; R[102] := R[92]; R[103] := R[93];

$$R_{101} := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$



$$R_{102} := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{103} := \begin{bmatrix} 5939 & 63664 \\ -799 & -8565 \end{bmatrix}$$

```
> R[111] := R[101]; R[112] := R[102];
> R[113] :=MatrixInverse(R[103]);
```

$$R_{111} := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$

$$R_{112} := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{113} := \begin{bmatrix} -8565 & -63664 \\ 799 & 5939 \end{bmatrix}$$

```
> R[121] := R[111]; R[122] := R[112]; R[123] := R[113].R[112]^3;
```

$$R_{121} := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$

$$R_{122} := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$R_{123} := \begin{bmatrix} -11 & 120 \\ 1 & -11 \end{bmatrix}$$

```
> T[1] := R[121]; T[2] := R[122]; T[3] := R[123];
```

$$T_1 := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$

$$T_2 := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$T_3 := \begin{bmatrix} -11 & 120 \\ 1 & -11 \end{bmatrix}$$

The participants calculate the matrices  $T_1, T_2, T_3$ , with the traces of these matrices they are able to reconstruct the correct secret.

## C.4. Example 6.2.3 executed with GAP

We give an example for a (3, 3)-secret sharing scheme. The finitely generated free group  $F$  with its free generating set  $X = \{a, b, c\}$  is defined in GAP:

```
LoadPackage("FGA");;
F:=FreeGroup("a", "b", "c");;
AssignGeneratorVariables(F);;
```

A Nielsen reduced set  $U = \{u_1, u_2, u_3\}$  with three elements is needed. The dealer chooses  $u_1 = b^2a$ ,  $u_2 = cab$  and  $u_3 = ac^{-1}b^{-1}a^3$ . Because of Theorem 4.2.13 the set  $U$  is a free generating set of a subgroup  $G$  of  $F$ . With the operation

```
▷ FreeGeneratorsOfGroup(G)
```

which returns a list of free Nielsen reduced generators, which defines a Nielsen reduced set of a finitely generated subgroup  $G$  of a free group, we can prove that  $U = \{b^2a, cab, ac^{-1}b^{-1}a^3\}$  is Nielsen reduced:

```
G:=Group (b^2*a, c*a*b, a*c^-1*b^-1*a^3);;

gap>FreeGeneratorsOfGroup(G);
[b^2*a, c*a*b, a*c^-1*b^-1*a^3]
```

The dealer calculates the Nielsen equivalent set  $V$  to  $U$  by applying the following Nielsen transformations:

$$\begin{aligned}
 (u_1, u_2, u_3) &\xrightarrow{(N1)_2} (u_1, u_2^{-1}, u_3) \\
 &\xrightarrow{(N2)_{1,3}} (u_1u_3, u_2^{-1}, u_3) \\
 &\xrightarrow{(N2)_{3,2}} (u_1u_3, u_2^{-1}, u_3u_2^{-1}) \\
 &\xrightarrow{(N2)_{2,3}} (u_1u_3, u_2^{-1}u_3u_2^{-1}, u_3u_2^{-1}) \\
 &\xrightarrow{(N1)_2} (u_1u_3, u_2u_3^{-1}u_2, u_3u_2^{-1}) \\
 &\xrightarrow{(N2)_{1,2}} (u_1u_3u_2u_3^{-1}u_2, u_2u_3^{-1}u_2, u_3u_2^{-1}) \\
 &\xrightarrow{(N1)_2} (u_1u_3u_2u_3^{-1}u_2, u_2^{-1}u_3u_2^{-1}, u_3u_2^{-1}) \\
 &\xrightarrow{(N2)_{3,2}} (u_1u_3u_2u_3^{-1}u_2, u_2^{-1}u_3u_2^{-1}, u_3u_2^{-2}u_3u_2^{-1}) \\
 &\xrightarrow{(N1)_1} (u_2^{-1}u_3u_2^{-1}u_3^{-1}u_1^{-1}, u_2^{-1}u_3u_2^{-1}, u_3u_2^{-2}u_3u_2^{-1}) \\
 &\xrightarrow{(N2)_{3,1}} (u_2^{-1}u_3u_2^{-1}u_3^{-1}u_1^{-1}, u_2^{-1}u_3u_2^{-1}, u_3u_2^{-2}u_3u_2^{-2}u_3u_2^{-1}u_3^{-1}u_1^{-1}) \\
 &\xrightarrow{(N1)_1} (u_1u_3u_2u_3^{-1}u_2, u_2^{-1}u_3u_2^{-1}, u_3u_2^{-2}u_3u_2^{-2}u_3u_2^{-1}u_3^{-1}u_1^{-1})
 \end{aligned}$$

In GAP define:

```
u_1:=b^2*a;;
u_2:=c*a*b;;
u_3:=a*c^-1*b^-1*a^3;;
```

```
v_1:=u_1*u_3*u_2*u_3^-1*u_2;;
v_2:=u_2^-1*u_3*u_2^-1;;
v_3:=u_3*u_2^-2*u_3*u_2^-2*u_3*u_2^-1*u_3^-1*u_1^-1;;
```

Write  $v_1$ ,  $v_2$  and  $v_3$  as words in  $X$ :

```
gap> v_1;
b^2*a^2*c^-1*b^-1*a^3*c*a*b*a^-3*b*c*a^-1*c*a*b
gap> v_2;
b^-1*a^-1*c^-1*a*c^-1*b^-1*a^3*b^-1*a^-1*c^-1
gap> v_3;
(a*c^-1*b^-1*a^3*(b^-1*a^-1*c^-1)^2)^2*a*c^-1*b^-1\
*a^3*b^-1*a^-1*c^-1*a^-3*b*c*a^-2*b^-2
```

Participant  $p_i$  gets the share  $v_i$ ,  $i = 1, 2, 3$ . If the participants combine their shares they obtain the set  $V$ .

If the set  $V = \{v_1, v_2, v_3\}$  of all shares is used as free generating set for a subgroup  $H$  of  $F$ , then the operation

▷ `FreeGeneratorsOfGroup(H)`

gives a Nielsen reduced generating set  $U'$  for  $H$ :

```
H:=Group(v_1,v_2,v_3);;

gap> H;
Group([ b^2*a^2*c^-1*b^-1*a^3*c*a*b*a^-3*b*c*a^-1*c*a*b,\
b^-1*a^-1*c^-1*a*c^-1*b^-1*a^3*b^-1*a^-1*c^-1,\
(a*c^-1*b^-1*a^3*(b^-1*a^-1*c^-1)^2)^2*a*c^-1*b^-1*a^3*b^-1\
*a^-1*c^-1*a^-3*b*c*a^-2*b^-2 ])
gap> FreeGeneratorsOfGroup(H);
[ b^2*a, c*a*b, a*c^-1*b^-1*a^3 ]
```

The participants get the Nielsen reduced generating set  $U' = \{b^2a, cab, ac^{-1}b^{-1}a^3\}$ , which is  $U$ , and hence they are able to reconstruct the correct secret

$$S = \sum_{i=1}^3 \frac{1}{|u'_i|_X} = \frac{1}{3} + \frac{1}{3} + \frac{1}{6} = \frac{5}{6}.$$

If just two participants combine their shares and generate Nielsen reduced sets, it is likely that no element in their Nielsen reduced set is of the length of one element in  $U$ . For example we can take a look at the free generating sets of subgroups which are just generated by two elements of the set  $V$ , see the following GAP-Code:

```
F1:=Group(v_1,v_2);;
F2:=Group(v_1,v_3);;
F3:=Group(v_2,v_3);;

gap> F1;
```

```

Group([ b^2*a^2*c^-1*b^-1*a^3*c*a*b*a^-3*b*c*a^-1*c*a*b,\
        b^-1*a^-1*c^-1*a*c^-1*b^-1*a^3*b^-1*a^-1*c^-1 ])
gap> FreeGeneratorsOfGroup(F1);
[ a^-3*b*c*a^-2*b^-2, c*a*b*a^-3*b*c*a^-1*c*a*b ]

gap> F2;
Group([ b^2*a^2*c^-1*b^-1*a^3*c*a*b*a^-3*b*c*a^-1*c*a*b,\
        (a*c^-1*b^-1*a^3*(b^-1*a^-1*c^-1)^2)^2*a*c^-1*b^-1*a^3*b^-1\
        *a^-1*c^-1*a^-3*b*c*a^-2*b^-2 ])
gap> FreeGeneratorsOfGroup(F2);
[ b^2*a^2*c^-1*b^-1*a^3*c*a*b*a^-3*b*c*a^-1*c*a*b,\
  c*a*b*a^-3*b*c*a^-1*(c*a*b)^2*a^-3*b*c*a^-1 ]

gap> F3;
Group([ b^-1*a^-1*c^-1*a*c^-1*b^-1*a^3*b^-1*a^-1*c^-1,\
        (a*c^-1*b^-1*a^3*(b^-1*a^-1*c^-1)^2)^2*a*c^-1*b^-1*a^3*b^-1\
        *a^-1*c^-1*a^-3*b*c*a^-2*b^-2 ])
gap> FreeGeneratorsOfGroup(F3);
[ c*a*b*a^-3*b*c*a^-1*c*a*b, b^2*a^2*c^-1*b^-1*a^3*\
  (c*a*b*a^-3*b*c*a^-1*c*a*b)^2*c*a*b*a^-3*b*c*a^-1 ]

```

## C.5. Example 7.0.7 executed with GAP

To execute the Example 7.0.7 the program GAP is used.

Firstly, Alice and Bob choose a free group  $F$  with free generating set  $X = \{a, b, c, d\}$ :

```
LoadPackage("FGA");;
F:=FreeGroup("a", "b", "c", "d");;
AssignGeneratorVariables(F);;
```

They use the set

$$\tilde{U} = \{ba^2, cd, d^2c^{-2}, a^{-1}b, a^4b^{-1}, b^3a^{-2}, bc^3, bc^{-1}bab^{-1}, c^2ba, c^2dab^{-1}, a^{-1}d^3c^{-1}, a^2db^2d^{-1}\}$$

as free generating set.

If this set is used as free generating set for a subgroup  $FU$  of  $F$ , then the operation

▷ `FreeGeneratorsOfGroup(FU)`

gives a Nielsen reduced generating set for  $FU$ , which is  $F_{\tilde{U}}$  in Example 7.0.7:

```
FU:=Group(b*a^2, c*d, d^2*c^-2, a^-1*b, a^4*b^-1, b^3*a^-2, b*c^3,\
          b*c^-1*b*a*b^-1, c^2*b*a, c^2*d*a*b^-1, a^-1*d^3*c^-1,\
          a^2*d*b^2*d^-1);;
```

```
gap> FU;
Group([ b*a^2, c*d, d^2*c^-2, a^-1*b, a^4*b^-1, b^3*a^-2,\
        b*c^3, b*c^-1*b*a*b^-1, c^2*b*a, c^2*d*a*b^-1,\
        a^-1*d^3*c^-1, a^2*d*b^2*d^-1 ])
gap> FreeGeneratorsOfGroup(FU);
[ b*a^2, c*d, d^2*c^-2, a^-1*b, a^4*b^-1, b^3*a^-2, b*c^3,\
  b*c^-1*b*a*b^-1, c^2*b*a, c^2*d*a*b^-1, a^-1*d^3*c^-1,\
  a^2*d*b^2*d^-1 ]
```

Secondly, they agree on the seed  $\overline{93}$  and to encrypt the message  $S = \text{ILIKEBOB}$  Alice needs 8 automorphisms, which are describable with Nielsen transformations as follows:

- Automorphism  $f_{x_1}$ :

$$\begin{aligned} (a, b, c, d) &\xrightarrow{(N1)_3} (a, b, c^{-1}, d) \\ &\xrightarrow{(N2)_{1.4}} (ad, b, c^{-1}, d) \\ &\xrightarrow{(N2)_{4.3}} (ad, b, c^{-1}, dc^{-1}) \\ &\xrightarrow{(N2)_{2.3}} (ad, bc^{-1}, c^{-1}, dc^{-1}) \\ &\xrightarrow{(N1)_3} (ad, bc^{-1}, c, dc^{-1}) \\ &\xrightarrow{(N2)_{1.4}} (ad^2c^{-1}, bc^{-1}, c, dc^{-1}) \\ &\xrightarrow{(N2)_{3.1}} (ad^2c^{-1}, bc^{-1}, cad^2c^{-1}, dc^{-1}) \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_1} : F &\rightarrow F \\
 a &\mapsto ad^2c^{-1}, \\
 b &\mapsto bc^{-1}, \\
 c &\mapsto cad^2c^{-1}, \\
 d &\mapsto dc^{-1}.
 \end{aligned}$$

- Automorphism  $f_{x_2}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N2)_{1,4}} (ad, b, c, d) \\
 &\xrightarrow{(N1)_2} (ad, b^{-1}, c, d) \\
 &\xrightarrow{(N2)_{2,4}} (ad, b^{-1}d, c, d) \\
 &\xrightarrow{(N2)_{3,1}} (ad, b^{-1}d, cad, d) \\
 &\xrightarrow{(N1)_2(N1)_1} (d^{-1}a^{-1}, d^{-1}b, cad, d) \\
 &\xrightarrow{(N2)_{1,3}} (d^{-1}a^{-1}cad, d^{-1}b, cad, d) \\
 &\xrightarrow{[(N2)_{4,3}]^2} (d^{-1}a^{-1}cad, d^{-1}b, cad, d(cad)^2) \\
 &\xrightarrow{(N1)_3} (d^{-1}a^{-1}cad, d^{-1}b, d^{-1}a^{-1}c^{-1}, d(cad)^2)
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_2} : F &\rightarrow F \\
 a &\mapsto d^{-1}a^{-1}cad, \\
 b &\mapsto d^{-1}b, \\
 c &\mapsto d^{-1}a^{-1}c^{-1}, \\
 d &\mapsto d(cad)^2.
 \end{aligned}$$

- Automorphism  $f_{x_3}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N1)_2} (a, b^{-1}, c, d) \\
 &\xrightarrow{(N2)_{4.2}} (a, b^{-1}, c, db^{-1}) \\
 &\xrightarrow{(N1)_4} (a, b^{-1}, c, bd^{-1}) \\
 &\xrightarrow{(N2)_{2.4}} (a, d^{-1}, c, bd^{-1}) \\
 &\xrightarrow{(N1)_2} (a, d, c, bd^{-1}) \\
 &\xrightarrow{(N2)_{4.2}} (a, d, c, b) \\
 &\xrightarrow{(N1)_3} (a, d, c^{-1}, b) \\
 &\xrightarrow{(N2)_{2.1}} (a, da, c^{-1}, b) \\
 &\xrightarrow{(N2)_{3.2}} (a, da, c^{-1}da, b) \\
 &\xrightarrow{[(N2)_{1.4}]^3} (ab^3, da, c^{-1}da, b) \\
 &\xrightarrow{(N1)_2} (ab^3, a^{-1}d^{-1}, c^{-1}da, b) \\
 &\xrightarrow{(N2)_{4.2}} (ab^3, a^{-1}d^{-1}, c^{-1}da, ba^{-1}d^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_3} : F &\rightarrow F \\
 a &\mapsto ab^3, \\
 b &\mapsto a^{-1}d^{-1}, \\
 c &\mapsto c^{-1}da, \\
 d &\mapsto ba^{-1}d^{-1}.
 \end{aligned}$$

- Automorphism  $f_{x_4}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{[(N2)_{3.1}]^2} (a, b, ca^2, d) \\
 &\xrightarrow{(N1)_2} (a, b^{-1}, ca^2, d) \\
 &\xrightarrow{[(N2)_{2.1}]^3} (a, b^{-1}a^3, ca^2, d) \\
 &\xrightarrow{(N2)_{2.4}} (a, b^{-1}a^3d, ca^2, d) \\
 &\xrightarrow{(N2)_{4.2}} (a, b^{-1}a^3d, ca^2, db^{-1}a^3d) \\
 &\xrightarrow{(N2)_{1.3}} (aca^2, b^{-1}a^3d, ca^2, db^{-1}a^3d)
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_4} : F &\rightarrow F \\
 a &\mapsto aca^2, \\
 b &\mapsto b^{-1}a^3d, \\
 c &\mapsto ca^2, \\
 d &\mapsto db^{-1}a^3d.
 \end{aligned}$$

- Automorphism  $f_{x_5}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N2)_{1,2}} (ab, b, c, d) \\
 &\xrightarrow{(N1)_3(N1)_1} (b^{-1}a^{-1}, b, c^{-1}, d) \\
 &\xrightarrow{[(N2)_{4,3}]^2} (b^{-1}a^{-1}, b, c^{-1}, dc^{-2}) \\
 &\xrightarrow{(N2)_{1,2}} (b^{-1}a^{-1}b, b, c^{-1}, dc^{-2}) \\
 &\xrightarrow{(N1)_2(N1)_3} (b^{-1}a^{-1}b, b^{-1}, c, dc^{-2}) \\
 &\xrightarrow{(N2)_{2,4}} (b^{-1}a^{-1}b, b^{-1}dc^{-2}, c, dc^{-2}) \\
 &\xrightarrow{(N2)_{3,1}} (b^{-1}a^{-1}b, b^{-1}dc^{-2}, cb^{-1}a^{-1}b, dc^{-2})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_5} : F &\rightarrow F \\
 a &\mapsto b^{-1}a^{-1}b, \\
 b &\mapsto b^{-1}dc^{-2}, \\
 c &\mapsto cb^{-1}a^{-1}b, \\
 d &\mapsto dc^{-2}.
 \end{aligned}$$

- Automorphism  $f_{x_6}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N1)_1} (a^{-1}, b, c, d) \\
 &\xrightarrow{(N2)_{2,3}} (a^{-1}, bc, c, d) \\
 &\xrightarrow{(N2)_{3,1}} (a^{-1}, bc, ca^{-1}, d) \\
 &\xrightarrow{(N1)_2} (a^{-1}, c^{-1}b^{-1}, ca^{-1}, d) \\
 &\xrightarrow{(N2)_{1,2}} (a^{-1}c^{-1}b^{-1}, c^{-1}b^{-1}, ca^{-1}, d) \\
 &\xrightarrow{(N2)_{4,2}} (a^{-1}c^{-1}b^{-1}, c^{-1}b^{-1}, ca^{-1}, dc^{-1}b^{-1})
 \end{aligned}$$



Hence, the automorphism is

$$\begin{aligned}
 f_{x_6} : F &\rightarrow F \\
 a &\mapsto a^{-1}c^{-1}b^{-1}, \\
 b &\mapsto c^{-1}b^{-1}, \\
 c &\mapsto ca^{-1}, \\
 d &\mapsto dc^{-1}b^{-1}.
 \end{aligned}$$

- Automorphism  $f_{x_7}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{[(N2)_{2,1}]^3} (a, ba^3, c, d) \\
 &\xrightarrow{(N1)_3} (a, ba^3, c^{-1}, d) \\
 &\xrightarrow{[(N2)_{4,3}]^3} (a, ba^3, c^{-1}, dc^{-3}) \\
 &\xrightarrow{(N1)_1} (a^{-1}, ba^3, c^{-1}, dc^{-3}) \\
 &\xrightarrow{(N2)_{1,2}} (a^{-1}ba^3, ba^3, c^{-1}, dc^{-3}) \\
 &\xrightarrow{(N1)_2} (a^{-1}ba^3, a^{-3}b^{-1}, c^{-1}, dc^{-3}) \\
 &\xrightarrow{(N2)_{2,4}} (a^{-1}ba^3, a^{-3}b^{-1}dc^{-3}, c^{-1}, dc^{-3}) \\
 &\xrightarrow{(N2)_{3,1}} (a^{-1}ba^3, a^{-3}b^{-1}dc^{-3}, c^{-1}a^{-1}ba^3, dc^{-3})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_7} : F &\rightarrow F \\
 a &\mapsto a^{-1}ba^3, \\
 b &\mapsto a^{-3}b^{-1}dc^{-3}, \\
 c &\mapsto c^{-1}a^{-1}ba^3, \\
 d &\mapsto dc^{-3}.
 \end{aligned}$$

- Automorphism  $f_{x_8}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N2)_{1,4}} (ad, b, c, d) \\
 &\xrightarrow{(N1)_2(N1)_3} (ad, b^{-1}, c^{-1}, d) \\
 &\xrightarrow{(N2)_{2,1}} (ad, b^{-1}ad, c^{-1}, d) \\
 &\xrightarrow{[(N2)_{3,4}]^2} (ad, b^{-1}ad, c^{-1}d^2, d) \\
 &\xrightarrow{(N1)_4(N1)_1(N1)_3} (d^{-1}a^{-1}, b^{-1}ad, d^{-2}c, d^{-1}) \\
 &\xrightarrow{(N2)_{4,2}} (d^{-1}a^{-1}, b^{-1}ad, d^{-2}c, d^{-1}b^{-1}ad)
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_8} : F &\rightarrow F \\
 a &\mapsto d^{-1}a^{-1}, \\
 b &\mapsto b^{-1}ad, \\
 c &\mapsto d^{-2}c, \\
 d &\mapsto d^{-1}b^{-1}ad.
 \end{aligned}$$

In GAP they define the automorphisms:

```

#f_{x1}
a1:=a*(d^2)*(c^(-1));;
b1:=b*(c^(-1));;
c1:=c*a*(d^2)*(c^(-1));;
d1:=d*(c^(-1));;
FF1:=Group(b1*a1^2, c1*d1, d1^2*c1^-2, a1^-1*b1, a1^4*b1^-1,\
           b1^3*a1^-2, b1*c1^3, b1*c1^-1*b1*a1*b1^-1, c1^2*b1*a1,\
           c1^2*d1*a1*b1^-1,a1^-1*d1^3*c1^-1, a1^2*d1*b1^2*d1^-1);;

#f_{x2}
a2:=d^-1*a^-1*c*a*d;;
b2:=d^-1*b;;
c2:=d^-1*a^-1*c^-1;;
d2:=d*(c*a*d)^2;;
FF2:=Group(b2*a2^2, c2*d2, d2^2*c2^-2, a2^-1*b2, a2^4*b2^-1,\
           b2^3*a2^-2, b2*c2^3, b2*c2^-1*b2*a2*b2^-1, c2^2*b2*a2,\
           c2^2*d2*a2*b2^-1,a2^-1*d2^3*c2^-1, a2^2*d2*b2^2*d2^-1);;

#f_{x3}
a3:=a*(b^3);;
b3:=(a^(-1))*(d^(-1));;
c3:=(c^(-1))*d*a;;
d3:=b*(a^(-1))*d^-1;;
FF3:=Group(b3*a3^2, c3*d3, d3^2*c3^-2, a3^-1*b3, a3^4*b3^-1,\
           b3^3*a3^-2, b3*c3^3, b3*c3^-1*b3*a3*b3^-1, c3^2*b3*a3,\
           c3^2*d3*a3*b3^-1,a3^-1*d3^3*c3^-1, a3^2*d3*b3^2*d3^-1);;

#f_{x4}
a4:=a*c*(a^2);;
b4:=(b^(-1))*(a^3)*d;;
c4:=c*(a^2);;
d4:=d*(b^(-1))*(a^3)*d;;
FF4:=Group(b4*a4^2, c4*d4, d4^2*c4^-2, a4^-1*b4, a4^4*b4^-1,\
           b4^3*a4^-2, b4*c4^3, b4*c4^-1*b4*a4*b4^-1, c4^2*b4*a4,\
           c4^2*d4*a4*b4^-1,a4^-1*d4^3*c4^-1, a4^2*d4*b4^2*d4^-1);;

#f_{x5}
a5:=b^-1*a^-1*b;;
b5:=b^-1*d*c^-2;;

```

```

c5:=c*b^-1*a^-1*b;;
d5:=d*c^-2;;
FF5:=Group(b5*a5^2, c5*d5, d5^2*c5^-2, a5^-1*b5, a5^4*b5^-1,\
           b5^3*a5^-2, b5*c5^3, b5*c5^-1*b5*a5*b5^-1, c5^2*b5*a5,\
           c5^2*d5*a5*b5^-1,a5^-1*d5^3*c5^-1, a5^2*d5*b5^2*d5^-1);;

#f_{x6}
a6:=(a^(-1))*(c^(-1))*(b^(-1));;
b6:=(c^(-1))*(b^(-1));;
c6:=c*(a^(-1));;
d6:=d*(c^(-1))*(b^(-1));;
FF6:=Group(b6*a6^2, c6*d6, d6^2*c6^-2, a6^-1*b6, a6^4*b6^-1,\
           b6^3*a6^-2, b6*c6^3, b6*c6^-1*b6*a6*b6^-1, c6^2*b6*a6,\
           c6^2*d6*a6*b6^-1,a6^-1*d6^3*c6^-1, a6^2*d6*b6^2*d6^-1);;

#f_{x7}
a7:=a^-1*b*a^3;;
b7:=a^-3*b^-1*d*c^-3;;
c7:=c^-1*a^-1*b*a^3;;
d7:=d*c^-3;;
FF7:=Group(b7*a7^2, c7*d7, d7^2*c7^-2, a7^-1*b7, a7^4*b7^-1,\
           b7^3*a7^-2, b7*c7^3, b7*c7^-1*b7*a7*b7^-1, c7^2*b7*a7,\
           c7^2*d7*a7*b7^-1,a7^-1*d7^3*c7^-1, a7^2*d7*b7^2*d7^-1);;

#f_{x8}
a8:=d^-1*a^-1;;
b8:=b^-1*a*d;;
c8:=d^-2*c;;
d8:=d^-1*b^-1*a*d;;
FF8:=Group(b8*a8^2, c8*d8, d8^2*c8^-2, a8^-1*b8, a8^4*b8^-1,\
           b8^3*a8^-2, b8*c8^3, b8*c8^-1*b8*a8*b8^-1, c8^2*b8*a8,\
           c8^2*d8*a8*b8^-1,a8^-1*d8^3*c8^-1, a8^2*d8*b8^2*d8^-1);;

```

Because of the one-to-one correspondence between the plaintext alphabet and the set  $U$  Alice gets for his message ILIKEBOB the ciphertext:

```

gap> FF1.3; FF2.8; FF3.3; FF4.9; FF5.2; FF6.11; FF7.4; FF8.11;
d*c^-1*d^-1*a^-1*d^-2*a^-1*c^-1
d^-1*b*c*a*b*d^-1*a^-1*c*a*d*b^-1*d
(b*a^-1*d^-1)^2*(a^-1*d^-1*c)^2
(c*a^2)^2*b^-1*a^3*d*a*c*a^2
c*b^-1*a^-1*b*d*c^-2
b*c*a*(d*c^-1*b^-1)^3*a*c^-1
a^-1*(a^-2*b^-1)^2*d*c^-3
(a*b^-1)^3*a*d*c^-1*d^2

```

Bob gets the entries for the tables, which are used for encryption, in GAP with:

```

gap> FF1.1;FF1.2;FF1.3;FF1.4;FF1.5;FF1.6;
b*(c^-1*a*d^2)^2*c^-1
c*a*d*(d*c^-1)^2

```

```

d*c^-1*d^-1*a^-1*d^-2*a^-1*c^-1
c*d^-2*a^-1*b*c^-1
(a*d^2*c^-1)^3*a*d^2*b^-1
(b*c^-1)^2*b*d^-2*a^-1*c*d^-2*a^-1
gap> FF1.7;FF1.8;FF1.9;FF1.10;FF1.11;FF1.12;
b*(a*d^2)^3*c^-1
b*d^-2*a^-1*c^-1*b*c^-1*a*d^2*b^-1
c*(a*d^2)^2*c^-1*b*c^-1*a*d^2*c^-1
c*(a*d^2)^2*c^-1*d*c^-1*a*d^2*b^-1
c*d^-2*a^-1*(d*c^-1)^2*d^-1*a^-1*c^-1
(a*d^2*c^-1)^2*d*(c^-1*b)^2*d^-1

gap> FF2.1;FF2.2;FF2.3;FF2.4;FF2.5;FF2.6;
d^-1*b*d^-1*a^-1*c^2*a*d
d^-1*a^-1*c^-1*(d*c*a)^2*d
((d*c*a)^2*d)^2*c*a*d*c*a*d
d^-1*a^-1*c^-1*a*b
d^-1*a^-1*c^4*a*d*b^-1*d
(d^-1*b)^3*d^-1*a^-1*c^-2*a*d
gap> FF2.7;FF2.8;FF2.9;FF2.10;FF2.11;FF2.12;
d^-1*b*(d^-1*a^-1*c^-1)^3
d^-1*b*c*a*b*d^-1*a^-1*c*a*d*b^-1*d
(d^-1*a^-1*c^-1)^2*d^-1*b*d^-1*a^-1*c*a*d
(d^-1*a^-1*c^-1)^2*d*c*a*d*c^2*a*d*b^-1*d
d^-1*a^-1*c^-1*(a*d^2*c*a*d*c)^3*a*d*c*a*d
d^-1*a^-1*c^2*a*d*(d*c*a)^2*b*d^-1*b*(d^-1*a^-1*c^-1)^2*d^-1

gap> FF3.1;FF3.2;FF3.3;FF3.4;FF3.5;FF3.6;
a^-1*d^-1*(a*b^3)^2
c^-1*d*a*b*a^-1*d^-1
(b*a^-1*d^-1)^2*(a^-1*d^-1*c)^2
b^-3*a^-2*d^-1
(a*b^3)^4*d*a
(a^-1*d^-1)^3*(b^-3*a^-1)^2
gap> FF3.7;FF3.8;FF3.9;FF3.10;FF3.11;FF3.12;
a^-1*d^-1*(c^-1*d*a)^3
(a^-1*d^-1)^2*c*a^-1*d^-1*a*b^3*d*a
c^-1*d*a*c^-1*a*b^3
(c^-1*d*a)^2*b*a^-1*d^-1*a*b^3*d*a
b^-3*a^-1*(b*a^-1*d^-1)^3*a^-1*d^-1*c
(a*b^3)^2*b*(a^-1*d^-1)^2*b^-1

gap> FF4.1;FF4.2;FF4.3;FF4.4;FF4.5;FF4.6;
b^-1*a^3*d*(a*c*a^2)^2
c*a^2*d*b^-1*a^3*d
(d*b^-1*a^3*d)^2*a^-2*c^-1*a^-2*c^-1
a^-2*c^-1*a^-1*b^-1*a^3*d
(a*c*a^2)^4*d^-1*a^-3*b
(b^-1*a^3*d)^3*a^-2*c^-1*a^-3*c^-1*a^-1
gap> FF4.7;FF4.8;FF4.9;FF4.10;FF4.11;FF4.12;

```

```

b^-1*a^3*d*(c*a^2)^3
b^-1*a^3*d*a^-2*c^-1*b^-1*a^3*d*a*c*a^2*d^-1*a^-3*b
(c*a^2)^2*b^-1*a^3*d*a*c*a^2
(c*a^2)^2*d*b^-1*a^3*d*a*c*a^2*d^-1*a^-3*b
a^-2*c^-1*a^-1*(d*b^-1*a^3*d)^3*a^-2*c^-1
a*c*a^3*c*(a^2*d*b^-1*a)^2*a^2

gap> FF5.1;FF5.2;FF5.3;FF5.4;FF5.5;FF5.6;
b^-1*d*c^-2*b^-1*a^-2*b
c*b^-1*a^-1*b*d*c^-2
d*c^-2*d*c^-1*(c^-1*b^-1*a*b)^2*c^-1
b^-1*a*d*c^-2
b^-1*a^-4*b*c^2*d^-1*b
(b^-1*d*c^-2)^3*b^-1*a^2*b
gap> FF5.7;FF5.8;FF5.9;FF5.10;FF5.11;FF5.12;
b^-1*d*c^-1*(b^-1*a^-1*b*c)^2*b^-1*a^-1*b
b^-1*d*c^-2*b^-1*a*b*c^-1*b^-1*d*c^-2*b^-1*a^-1*b*c^2*d^-1*b
c*b^-1*a^-1*b*c*b^-1*a^-1*d*c^-2*b^-1*a^-1*b
(c*b^-1*a^-1*b)^2*d*c^-2*b^-1*a^-1*b*c^2*d^-1*b
b^-1*a*b*(d*c^-2)^3*b^-1*a*b*c^-1
b^-1*a^-2*b*(d*c^-2*b^-1)^2

gap> FF6.1;FF6.2;FF6.3;FF6.4;FF6.5;FF6.6;
(c^-1*b^-1*a^-1)^2*c^-1*b^-1
c*a^-1*d*c^-1*b^-1
(d*c^-1*b^-1)^2*a*c^-1*a*c^-1
b*c*a*c^-1*b^-1
(a^-1*c^-1*b^-1)^3*a^-1
(c^-1*b^-1)^2*a*b*c*a
gap> FF6.7;FF6.8;FF6.9;FF6.10;FF6.11;FF6.12;
c^-1*b^-1*(c*a^-1)^3
c^-1*b^-1*a*c^-2*b^-1*a^-1
c*a^-1*c*(a^-1*c^-1*b^-1)^2
(c*a^-1)^2*d*c^-1*b^-1*a^-1
b*c*a*(d*c^-1*b^-1)^3*a*c^-1
(a^-1*c^-1*b^-1)^2*d*(c^-1*b^-1)^2*d^-1

gap> FF7.1;FF7.2;FF7.3;FF7.4;FF7.5;FF7.6;
a^-3*b^-1*d*c^-3*a^-1*(b*a^2)^2*a
c^-1*a^-1*b*a^3*d*c^-3
d*c^-3*d*c^-3*(a^-3*b^-1*a*c)^2
a^-1*(a^-2*b^-1)^2*d*c^-3
a^-1*(b*a^2)^4*a*c^3*d^-1*b*a^3
(a^-3*b^-1*d*c^-3)^3*a^-1*(a^-2*b^-1)^2*a
gap> FF7.7;FF7.8;FF7.9;FF7.10;FF7.11;FF7.12;
a^-3*b^-1*d*c^-3*(c^-1*a^-1*b*a^3)^3
a^-3*b^-1*d*c^-3*a^-3*b^-1*a*c*a^-3*b^-1*d*c^-3*a^-1*b*a^3*c^3*d^-1*b*a^3
c^-1*a^-1*b*a^3*c^-1*a^-1*d*c^-3*a^-1*b*a^3
(c^-1*a^-1*b*a^3)^2*d*c^-3*a^-1*b*a^3*c^3*d^-1*b*a^3
a^-3*b^-1*a*(d*c^-3)^3*a^-3*b^-1*a*c

```

```

a^-1*(b*a^2)^2*a*(d*c^-3*a^-3*b^-1)^2

gap> FF8.1;FF8.2;FF8.3;FF8.4;FF8.5;FF8.6;
b^-1*d^-1*a^-1
d^-2*c*d^-1*b^-1*a*d
d^-1*(b^-1*a)^2*(d*c^-1*d)^2*d
a*d*b^-1*a*d
(d^-1*a^-1)^5*b
(b^-1*a*d)^3*a*d*a*d
gap> FF8.7;FF8.8;FF8.9;FF8.10;FF8.11;FF8.12;
b^-1*a*(d^-1*c*d^-1)^2*d^-1*c
b^-1*a*d*c^-1*d^2*b^-1*d^-1*a^-1*b
(d^-2*c)^2*b^-1
(d^-2*c)^2*d^-1*b^-1*d^-1*a^-1*b
(a*b^-1)^3*a*d*c^-1*d^2
(d^-1*a^-1)^2*d^-1*(b^-1*a*d)^2*d

```

## C.6. Example 7.2.4 calculations in Maple 16 and GAP

In Example 7.2.4, the Example 7.0.7 is extended with a faithful representation, thus the ciphertext is a sequence of matrices in  $SL(2, \mathbb{Q})$ .

We first prove with GAP, that the set  $M = \{X_1X_2, X_3X_1^2, X_2X_3X_2, X_1^{-1}X_2\}$  is a free generating set for a subgroup  $F_\varphi$  of rank 4 of  $F_{\varphi_1} = \langle X_1, X_2, X_3 \mid \ \rangle$ :

```
LoadPackage("FGA");;
Fphi1:=FreeGroup("X1", "X2", "X3");;
AssignGeneratorVariables(Fphi1);;

Fphi:=Group(X1*X2, X3*X1^2, X2*X3*X2, X1^-1*X2);;

gap> Rank(Fphi);
4
```

We show now the calculations, which are needed for this example and were executed with the program Maple 16:

```
> restart; with(LinearAlgebra):
```

For the faithful representation  $\varphi$  they need matrices in  $SL(2, \mathbb{Q})$  and generate them with Theorem 4.2.18. For this they choose rational numbers  $r_i =: r[i]$ ,  $1 \leq i \leq 3$ , with the properties (4.1). Hence, these rational numbers were chosen as follows and the inequalities (4.1) were proved:

```
> r[1] := 7/2;
> r[2] := 15/2;
> r[3] := 23/2;
```

$$r_1 := \frac{7}{2}$$

$$r_2 := \frac{15}{2}$$

$$r_3 := \frac{23}{2}$$

```
> r[1]-2;
> r[2]-r[1]-3;
> r[3]-r[2]-3;
```

$$\frac{3}{2}$$

$$1$$

$$1$$

All results are greater than 0, hence they can generate with the numbers  $r_1$ ,  $r_2$  and  $r_3$  matrices which generate a free subgroup of  $SL(2, \mathbb{Q})$  of rank 3. The matrices for Alice and Bob are:

```
> X[1] := Matrix([[ -r[1], r[1]^2-1], [1, -r[1]]]);
> X[2] := Matrix([[ -r[2], r[2]^2-1], [1, -r[2]]]);
> X[3] := Matrix([[ -r[3], r[3]^2-1], [1, -r[3]]]);
```

$$X_1 := \begin{bmatrix} -\frac{7}{2} & \frac{45}{4} \\ 1 & -\frac{7}{2} \end{bmatrix}$$

$$X_2 := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$X_3 := \begin{bmatrix} \frac{-23}{2} & \frac{525}{4} \\ 1 & \frac{-23}{2} \end{bmatrix}$$

Because of the free generating set  $\{X_1X_2, X_3X_1^2, X_2X_3X_2, X_1^{-1}X_2\}$  and the chosen faithful representation  $\varphi$ , they define in Maple:

```
> a:=X[1].X[2];
> b:=X[3].X[1].X[1];
> c:=X[2].X[3].X[2];
> d:=MatrixInverse(X[1]).X[2];
```

$$a := \begin{bmatrix} \frac{75}{2} & \frac{-1111}{4} \\ -11 & \frac{163}{2} \end{bmatrix}$$

$$b := \begin{bmatrix} -1189 & 3990 \\ 104 & -349 \end{bmatrix}$$

$$c := \begin{bmatrix} -2681 & 19966 \\ 360 & -2681 \end{bmatrix}$$

$$d := \begin{bmatrix} 15 & -109 \\ 4 & -29 \end{bmatrix}$$

The ciphertext  $C' = C_1C_2C_3C_4C_5C_6C_7C_8$ , as sequence of matrices in  $SL(2, \mathbb{Q})$ , is now

```
> C[1]:=d.MatrixInverse(c).MatrixInverse(d).MatrixInverse(a).MatrixInverse(d).MatrixInverse(d).MatrixInverse(a).MatrixInverse(c);
> C[2]:=MatrixInverse(d).b.c.a.b.MatrixInverse(d).MatrixInverse(a).c.a.d.MatrixInverse(b).d;
> C[3]:=b.MatrixInverse(a).MatrixInverse(d).b.MatrixInverse(a).MatrixInverse(d).MatrixInverse(a).MatrixInverse(d).c.MatrixInverse(a).MatrixInverse(d).c;
> C[4]:=c.a.a.c.a.a.MatrixInverse(b).a.a.a.d.a.c.a.a;
> C[5]:=c.MatrixInverse(b).MatrixInverse(a).b.d.MatrixInverse(c).MatrixInverse(c);
> C[6]:=b.c.a.d.MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(b).a.MatrixInverse(c);
> C[7]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c);
> C[8]:=a.MatrixInverse(b).a.MatrixInverse(b).a.MatrixInverse(b).a.d.MatrixInverse(c).d.d;
```

$$C_1 := \begin{bmatrix} \frac{-42974309355909}{2} & \frac{-6400784021410159}{4} \\ -62588240305379 & \frac{-932216979117085}{2} \end{bmatrix}$$

$$C_2 := \begin{bmatrix} \frac{-3240070331754423030683243991}{2} & \frac{47007695458416827592369656315}{4} \\ -223326322203710575272321977 & \frac{3240070327830150751386194361}{2} \end{bmatrix}$$



$$C_3 := \begin{bmatrix} \frac{-6899014060703475554169965}{2} & \frac{102756972145191520348785607}{4} \\ 301722468685102729969483 & \frac{-4493988131847945704997109}{2} \end{bmatrix}$$

$$C_4 := \left[ \frac{-397074726172421275253684843812134445}{2}, \right. \\ \left. \frac{5883318761059670223751985896578473377}{4} \right]$$

$$\left[ \frac{26659253089426526822952736194350493}{2}, \right. \\ \left. \frac{-395000924306510751052288425218790757}{2} \right]$$

$$C_5 := \begin{bmatrix} \frac{46475888407425825}{2} & \frac{692232489736400389}{4} \\ -3120351373297111 & \frac{-46475896943687759}{2} \end{bmatrix}$$

$$C_6 := \begin{bmatrix} \frac{-37154085868492177463035768197599}{2} & \frac{-553374013794643763898030444104547}{4} \\ 1624906569753714749910956723073 & \frac{24201404758781402065719318991873}{2} \end{bmatrix}$$

$$C_7 := \begin{bmatrix} \frac{-3418963163764785449276501363}{2} & \frac{-50923553357916815212095363641}{4} \\ -230751369629481141540301125 & \frac{-3436913216344813651054341083}{2} \end{bmatrix}$$

$$C_8 := \begin{bmatrix} \frac{2739747352948144349387}{2} & \frac{-39628644296581967709615}{4} \\ -402070084312200114547 & \frac{5815679440792026855107}{2} \end{bmatrix}$$

Decryption:

If Bob gets the above ciphertext as sequence of matrices he has to calculate a table like Table 7.7 (page 172). Therefore, he first defines in Maple the common matrices  $X_1$ ,  $X_2$  and  $X_3$ :

```
> restart;
> with(LinearAlgebra):
> r[1] := 7/2;
> r[2] := 15/2;
> r[3] := 23/2;
```

$$r_1 := \frac{7}{2}$$

$$r_2 := \frac{15}{2}$$

$$r_3 := \frac{23}{2}$$

```
> X[1] := Matrix([[ -r[1], r[1]^2-1], [1, -r[1]]]);
> X[2] := Matrix([[ -r[2], r[2]^2-1], [1, -r[2]]]);
> X[3] := Matrix([[ -r[3], r[3]^2-1], [1, -r[3]]]);
```

$$X_1 := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$

$$X_2 := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$X_3 := \begin{bmatrix} \frac{-23}{2} & \frac{525}{4} \\ 1 & \frac{-23}{2} \end{bmatrix}$$

Because of the free generating set  $\{X_1X_2, X_3X_1^2, X_2X_3X_2, X_1^{-1}X_2\}$  and the chosen faithful representation  $\varphi$ , he defines in Maple:

```
> a:=X[1].X[2];
> b:=X[3].X[1].X[1];
> c:=X[2].X[3].X[2];
> d:=MatrixInverse(X[1]).X[2];
```

$$a := \begin{bmatrix} \frac{75}{2} & \frac{-1111}{4} \\ -11 & \frac{163}{2} \end{bmatrix}$$

$$b := \begin{bmatrix} -1189 & 3990 \\ 104 & -349 \end{bmatrix}$$

$$c := \begin{bmatrix} -2681 & 19966 \\ 360 & -2681 \end{bmatrix}$$

$$d := \begin{bmatrix} 15 & -109 \\ 4 & -29 \end{bmatrix}$$

With these definitions he is able to calculate the required Table C.1 (page 290) with which he is able to decrypt the ciphertext  $C'$ .

Table C.1.: Plaintext alphabet  $A = \{A, E, I, O, U, T, M, L, K, Y, B, S\}$  corresponding to ciphertext alphabet  $U_{\varphi(f_{x_i})}$  depending on the automorphisms  $f_{x_i}$  and the faithful representation  $\varphi$

	$U_{\varphi(f_{x_1})}$	$U_{\varphi(f_{x_2})}$	$\dots$	$U_{\varphi(f_{x_8})}$
$a_1 = A$	$\varphi(f_{x_1}(u_1)) = N_{11}$	$\varphi(f_{x_2}(u_1)) = N_{21}$	$\dots$	$\varphi(f_{x_8}(u_1)) = N_{81}$
$a_2 = E$	$\varphi(f_{x_1}(u_2)) = N_{12}$	$\varphi(f_{x_2}(u_2)) = N_{22}$	$\dots$	$\varphi(f_{x_8}(u_2)) = N_{82}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{12} = S$	$\varphi(f_{x_1}(u_{12})) = N_{112}$	$\varphi(f_{x_2}(u_{12})) = N_{212}$	$\dots$	$\varphi(f_{x_8}(u_{12})) = N_{812}$

For Table C.1 (page 290) he first calculates the entries in the column  $U_{\varphi(f_{x_1})}$ :

```

> N[11]:=b.MatrixInverse(c).a.d.d.MatrixInverse(c).a.d.d.
> MatrixInverse(c);
> N[12]:=c.a.d.d.MatrixInverse(c).d.MatrixInverse(c);
> N[13]:=d.MatrixInverse(c).MatrixInverse(d).MatrixInverse(a).MatrixInve
> rse(d).MatrixInverse(d).MatrixInverse(a).MatrixInverse(c);
> N[14]:=c.MatrixInverse(d).MatrixInverse(d).MatrixInverse(a).b.MatrixIn
> verse(c);
> N[15]:=a.d.d.MatrixInverse(c).a.d.d.MatrixInverse(c).a.d.d.MatrixInver
> se(c).a.d.d.MatrixInverse(b);
> N[16]:=b.MatrixInverse(c).b.MatrixInverse(c).b.MatrixInverse(d).Matrix
> Inverse(d).MatrixInverse(a).c.MatrixInverse(d).MatrixInverse(d).Matrix
> Inverse(a);
> N[17]:=b.a.d.d.a.d.d.a.d.d.MatrixInverse(c);
> N[18]:=b.MatrixInverse(d).MatrixInverse(d).MatrixInverse(a).MatrixInve
> rse(c).b.MatrixInverse(c).a.d.d.MatrixInverse(b);
> N[19]:=c.a.d.d.a.d.d.MatrixInverse(c).b.MatrixInverse(c).a.d.d.MatrixI
> nverse(c);
> N[110]:=c.a.d.d.a.d.d.MatrixInverse(c).d.MatrixInverse(c).a.d.d.Matrix
> Inverse(b);
> N[111]:=c.MatrixInverse(d).MatrixInverse(d).MatrixInverse(a).d.MatrixI
> nverse(c).d.MatrixInverse(c).MatrixInverse(d).MatrixInverse(a).MatrixI
> nverse(c);
> N[112]:=a.d.d.MatrixInverse(c).a.d.d.MatrixInverse(c).d.MatrixInverse(c)
> .b.MatrixInverse(c).b.MatrixInverse(d);

```

$$N_{11} := \begin{bmatrix} \frac{1726044910473446627}{2} & \frac{25708356233079285239}{4} \\ -75487096670467781 & \frac{-1124332953580545765}{2} \end{bmatrix}$$

$$N_{12} := \begin{bmatrix} \frac{-31245270331619}{2} & \frac{-465378427039713}{4} \\ 2097780379171 & \frac{31245104387701}{2} \end{bmatrix}$$

$$N_{13} := \begin{bmatrix} \frac{-429743093559909}{2} & \frac{-6400784021410159}{4} \\ -62588240305379 & \frac{-932216979117085}{2} \end{bmatrix}$$

$$N_{14} := \begin{bmatrix} \frac{3181531972845}{2} & \frac{47387139217729}{4} \\ -213606308371 & \frac{-3181546487339}{2} \end{bmatrix}$$

$$N_{15} := \begin{bmatrix} \frac{2131284812621970980298075}{2} & \frac{48732748634630267751942089}{4} \\ -312796606586025044761211 & \frac{-7152229637375004077469917}{2} \end{bmatrix}$$

$$N_{16} := \begin{bmatrix} \frac{-5409749175019023063569477}{2} & \frac{-36860106580604442519169935}{4} \\ 236590749441386152662525 & \frac{1612045209168853659502723}{2} \end{bmatrix}$$

$$N_{17} := \begin{bmatrix} 795521381235160141 & 5924395980590698676 \\ -69583114032935312 & -518198417814550971 \end{bmatrix}$$

$$\begin{aligned}
 N_{18} &:= \begin{bmatrix} -90099029000979888807221 & -1030076624795949617586074 \\ 7880806953101379022888 & 90099029000976615232691 \end{bmatrix} \\
 N_{19} &:= \begin{bmatrix} 51707261346868077665909739 & 385073611490674350228100114 \\ -6943161455342564033343592 & -51707017295619157216197933 \end{bmatrix} \\
 N_{110} &:= \begin{bmatrix} 1362002520154399003411251 & 15571388221164541516505605 \\ -182887338329092260567748 & -2090899028244770708376289 \end{bmatrix} \\
 N_{111} &:= \begin{bmatrix} \frac{143452020684119915871}{2} & \frac{2136637948565376806965}{4} \\ -9631289840302078855 & \frac{-143452697761124185201}{2} \end{bmatrix} \\
 N_{112} &:= \begin{bmatrix} \frac{10122094452395075481217127}{2} & \frac{-76240613192995084024349669}{4} \\ -1485562500845350307519449 & \frac{11189403194530487717235255}{2} \end{bmatrix}
 \end{aligned}$$

Second, he calculates the entries in the column  $U_{\varphi(f_{x_2})}$ :

```

> N[21]:=MatrixInverse(d).b.MatrixInverse(d).MatrixInverse(a).c.c.a.d;
> N[22]:=MatrixInverse(d).MatrixInverse(a).MatrixInverse(c).d.c.a.d.c.a.
> d;
> N[23]:=d.c.a.d.c.a.d.d.c.a.d.c.a.d.c.a.d.c.a.d;
> N[24]:=MatrixInverse(d).MatrixInverse(a).MatrixInverse(c).a.b;
> N[25]:=MatrixInverse(d).MatrixInverse(a).c.c.c.c.a.d.MatrixInverse(b).
> d;
> N[26]:=MatrixInverse(d).b.MatrixInverse(d).b.MatrixInverse(d).b.Matrix
> Inverse(d).MatrixInverse(a).MatrixInverse(c).MatrixInverse(c).a.d;
> N[27]:=MatrixInverse(d).b.MatrixInverse(d).MatrixInverse(a).MatrixInve
> rse(c).MatrixInverse(d).MatrixInverse(a).MatrixInverse(c).MatrixInvers
> e(d).MatrixInverse(a).MatrixInverse(c);
> N[28]:=MatrixInverse(d).b.c.a.b.MatrixInverse(d).MatrixInverse(a).c.a.
> d.MatrixInverse(b).d;
> N[29]:=MatrixInverse(d).MatrixInverse(a).MatrixInverse(c).MatrixInvers
> e(d).MatrixInverse(a).MatrixInverse(c).MatrixInverse(d).b.MatrixInvers
> e(d).MatrixInverse(a).c.a.d;
> N[210]:=MatrixInverse(d).MatrixInverse(a).MatrixInverse(c).MatrixInver
> se(d).MatrixInverse(a).MatrixInverse(c).d.c.a.d.c.c.a.d.MatrixInverse(
> b).d;
> N[211]:=MatrixInverse(d).MatrixInverse(a).MatrixInverse(c).a.d.d.c.a.d
> .c.a.d.d.c.a.d.c.a.d.d.c.a.d.c.a.d.c.a.d;
> N[212]:=MatrixInverse(d).MatrixInverse(a).c.c.a.d.d.c.a.d.c.a.b.Matrix
> Inverse(d).b.MatrixInverse(d).MatrixInverse(a).MatrixInverse(c).Matrix
> Inverse(d).MatrixInverse(a).MatrixInverse(c).MatrixInverse(d);

```

$$\begin{aligned}
 N_{21} &:= \begin{bmatrix} 389795903484122413 & -2819360910238709533 \\ 53734441988781860 & -388656689109694783 \end{bmatrix} \\
 N_{22} &:= \begin{bmatrix} \frac{34818098324541438944473}{2} & \frac{-503672740055148162378181}{4} \\ 2406919959187600869799 & \frac{-34818098324541696255255}{2} \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 N_{23} &:= \\
 &[-129947831392256654391379918549839222145036327, \\
 &939901709902249703742555386199933409489204602] \\
 &[-34613503284060659775514069446954072249786856, \\
 &250356551347068858962670784732660569304552793] \\
 N_{24} &:= \begin{bmatrix} 396104100073 & -1329232007756 \\ 54764097424 & -183775404391 \end{bmatrix} \\
 N_{25} &:= \begin{bmatrix} 11174267271911668375467101 & -81059436849408272206936373 \\ 1544915977518850788019012 & -11207000519140712027900975 \end{bmatrix} \\
 N_{26} &:= \begin{bmatrix} -236310193323463285933886391 & 1709214803716582701013996097 \\ -32576012885043522519239540 & 235619981881035997967538469 \end{bmatrix} \\
 N_{27} &:= \begin{bmatrix} 9531492161404806907617847 & 70983133459497417507196019 \\ 1313942522318989432917316 & 9785221016866591539405915 \end{bmatrix} \\
 N_{28} &:= \begin{bmatrix} \frac{-3240070331754423030683243991}{2} & \frac{47007695458416827592369656315}{4} \\ -223326322203710575272321977 & \frac{3240070327830150751386194361}{2} \end{bmatrix} \\
 N_{29} &:= \begin{bmatrix} \frac{-12382178550188482311769807597}{2} & \frac{179118507280143386620295182193}{4} \\ -855960380514708613164396211 & \frac{12382178550187706387285804827}{2} \end{bmatrix} \\
 N_{210} &:= [-21110929144428898215300010362223029327, \\
 &153141139922135745238345648793303290342] \\
 &[-2918730152410756047224184025644787864, \\
 &21172808624733035641035893652532742081] \\
 N_{211} &:= \left[ \frac{-91706563164184834841532011018350101065765948680031435964924973}{2}, \right. \\
 &\left. \frac{1326611681068995502652878382969282252045951186401367136911128377}{4} \right] \\
 &\left[ \frac{-6339529379547013672054019614913843728524362886626126071455963,}{2} \right. \\
 &\left. \frac{91706563164190581596008000085122503931656189978277573261459739}{2} \right] \\
 N_{212} &:= [82335301850873934413508015209229035820945924887161, \\
 &-309107513178964843246713866147639639578017068212624] \\
 &[11383397250841957747919103394908684555473783833984, \\
 &-42736147638215940177555361818088748311302543810615]
 \end{aligned}$$

Third, he calculates the entries in the column  $U_{\varphi(f_{x_3})}$ :

```

> N[31]:=MatrixInverse(a).MatrixInverse(d).a.b.b.b.a.b.b.b;
> N[32]:=MatrixInverse(c).d.a.b.MatrixInverse(a).MatrixInverse(d);
> N[33]:=b.MatrixInverse(a).MatrixInverse(d).b.MatrixInverse(a).MatrixIn
> verse(d).MatrixInverse(a).MatrixInverse(d).c.MatrixInverse(a).MatrixIn
> verse(d).c;
> N[34]:=MatrixInverse(b).MatrixInverse(b).MatrixInverse(b).MatrixInvers
> e(a).MatrixInverse(a).MatrixInverse(d);
> N[35]:=a.b.b.b.a.b.b.b.a.b.b.b.a.b.b.b.d.a;
> N[36]:=MatrixInverse(a).MatrixInverse(d).MatrixInverse(a).MatrixInvers
> e(d).MatrixInverse(a).MatrixInverse(d).MatrixInverse(b).MatrixInverse(
> b).MatrixInverse(b).MatrixInverse(a).MatrixInverse(b).MatrixInverse(b)
> .MatrixInverse(b).MatrixInverse(a);
> N[37]:=MatrixInverse(a).MatrixInverse(d).MatrixInverse(c).d.a.MatrixIn
> verse(c).d.a.MatrixInverse(c).d.a;
> N[38]:=MatrixInverse(a).MatrixInverse(d).MatrixInverse(a).MatrixInvers
> e(d).c.MatrixInverse(a).MatrixInverse(d).a.b.b.b.d.a;
> N[39]:=MatrixInverse(c).d.a.MatrixInverse(c).a.b.b.b;
> N[310]:=MatrixInverse(c).d.a.MatrixInverse(c).d.a.b.MatrixInverse(a).M
> atrixInverse(d).a.b.b.b.d.a;
> N[311]:=MatrixInverse(b).MatrixInverse(b).MatrixInverse(b).MatrixInvers
> e(a).b.MatrixInverse(a).MatrixInverse(d).b.MatrixInverse(a).MatrixInve
> rse(d).b.MatrixInverse(a).MatrixInverse(d).MatrixInverse(a).MatrixInve
> rse(d).c;
> N[312]:=a.b.b.b.a.b.b.b.b.MatrixInverse(a).MatrixInverse(d).MatrixInve
> rse(a).MatrixInverse(d).MatrixInverse(b);

```

$$N_{31} := \begin{bmatrix} \frac{-875508157155713045018858865}{2} & \frac{5875995723000494713919774509}{4} \\ -59089551091571537879850991 & \frac{396581055984771055113986671}{2} \end{bmatrix}$$

$$N_{32} := \begin{bmatrix} -52462153571035 & 197040748651696 \\ -7044527054176 & 26458290218157 \end{bmatrix}$$

$$N_{33} := \begin{bmatrix} \frac{-6899014060703475554169965}{2} & \frac{102756972145191520348785607}{4} \\ 301722468685102729969483 & \frac{-4493988131847945704997109}{2} \end{bmatrix}$$

$$N_{34} := \begin{bmatrix} \frac{1736041194186217}{2} & \frac{-13040669107719885}{4} \\ 258665645503951 & \frac{-1943025951139943}{2} \end{bmatrix}$$

$$N_{35} := \left[ \frac{2676005265677868729323807778096848696881970346991}{2}, \right. \\ \left. \frac{-39649443737453961182911866890506362179673136614829}{4} \right]$$

$$\left[ \frac{-392531276610950108107075857748690863246534068369,}{2}, \right. \\ \left. \frac{5816000053062062920287043877980031608507919293455}{2} \right]$$

$$N_{36} := \left[ \begin{array}{c} \frac{143469085237161269143044743153795}{2} \quad \frac{978072450356992870819485910375329}{4} \\ 9682977735144453927523988429869 \quad \frac{66011808359337462336418813538459}{2} \end{array} \right]$$

$$N_{37} := \left[ \begin{array}{c} \frac{262766997214803713261}{2} \quad \frac{-3893321884154026856671}{4} \\ 17734600975882186301 \quad \frac{-262767055292328578747}{2} \end{array} \right]$$

$$N_{38} := \left[ \begin{array}{c} \frac{60046959286768006113664103}{2} \quad \frac{-889695010836535207325903853}{4} \\ 4052673569889420773596575 \quad \frac{-60047061475005452764043257}{2} \end{array} \right]$$

$$N_{39} := \left[ \begin{array}{c} \frac{81250457122162677557}{2} \quad \frac{-545314552057052265903}{4} \\ 5455085279589136317 \quad \frac{-36611946455877241171}{2} \end{array} \right]$$

$$N_{310} := \left[ \begin{array}{c} -264642814125471122620337910440849 \quad 1960560486141522671648480208507617 \\ -35535778402189873460069830975764 \quad 263260664038220168770908609864163 \end{array} \right]$$

$$N_{311} := \left[ \begin{array}{c} \frac{-33799498112481785080551313536439331923}{2}, \\ \frac{503424700328119525649401721213732913751}{4} \end{array} \right]$$

$$\left[ \begin{array}{c} -5036037754318909623407973468834742789, \\ \frac{75008977614754793145630085595903685045}{2} \end{array} \right]$$

$$N_{312} := [-12489517190626465361408670713063501, \\ -142791693237797324141014980979372933] \\ [3664063139165114201184457628059780, \\ 41890953171852949920484820282758239]$$

Fourth, he calculates the entries in the column  $U_{\varphi(f_{x_4})}$ :

```

> N[41]:=MatrixInverse(b).a.a.a.d.a.c.a.a.a.c.a.a;
> N[42]:=c.a.a.d.MatrixInverse(b).a.a.a.d;
> N[43]:=d.MatrixInverse(b).a.a.a.d.d.MatrixInverse(b).a.a.a.d.MatrixInv
> erse(a).MatrixInverse(a).MatrixInverse(c).MatrixInverse(a).MatrixInver
> se(a).MatrixInverse(c);
> N[44]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(c).MatrixInvers
> e(a).MatrixInverse(b).a.a.a.d;
> N[45]:=a.c.a.a.a.c.a.a.a.c.a.a.a.c.a.a.MatrixInverse(d).MatrixInverse(
> a).MatrixInverse(a).MatrixInverse(a).b;
> N[46]:=MatrixInverse(b).a.a.a.d.MatrixInverse(b).a.a.a.d.MatrixInverse
> (b).a.a.a.d.MatrixInverse(a).MatrixInverse(a).MatrixInverse(c).MatrixI
> nverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInverse(c).MatrixInv
> erse(a);
> N[47]:=MatrixInverse(b).a.a.a.d.c.a.a.c.a.a.c.a.a;
> N[48]:=MatrixInverse(b).a.a.a.d.MatrixInverse(a).MatrixInverse(a).Matr
> ixInverse(c).MatrixInverse(b).a.a.a.d.a.c.a.a.MatrixInverse(d).MatrixI
> nverse(a).MatrixInverse(a).MatrixInverse(a).b;
> N[49]:=c.a.a.c.a.a.MatrixInverse(b).a.a.a.d.a.c.a.a;
> N[410]:=c.a.a.c.a.a.d.MatrixInverse(b).a.a.a.d.a.c.a.a.MatrixInverse(d
> ).MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).b;
> N[411]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(c).MatrixInver
> se(a).d.MatrixInverse(b).a.a.a.d.d.MatrixInverse(b).a.a.a.d.d.MatrixIn
> verse(b).a.a.a.d.MatrixInverse(a).MatrixInverse(a).MatrixInverse(c);
> N[412]:=a.c.a.a.a.c.a.a.d.MatrixInverse(b).a.a.a.d.MatrixInverse(b).a.
> a.a;

```

$$N_{41} := \begin{bmatrix} -513488210929547673539574533663 & 3804088532737489619819457791461 \\ -153017069607382517165666970852 & 1133600474980335755589237657517 \end{bmatrix}$$

$$N_{42} := \begin{bmatrix} \frac{8222742539238989319}{2} & \frac{-118948807013625144725}{4} \\ -552067809895253689 & \frac{7986119845574387591}{2} \end{bmatrix}$$

$$N_{43} := \begin{bmatrix} \frac{-1563284215014946800339122756440950793}{2}, \\ \frac{-23284247669073816353945187259984437557}{4} \\ \frac{-207549619755460346794126167987042697,}{2} \\ \frac{-3091335985863599924019109669659549081}{2} \end{bmatrix}$$

$$N_{44} := \begin{bmatrix} 79686112408670975985 & -576362932521190220726 \\ 10756289961320657336 & -77799338401159316671 \end{bmatrix}$$

$$N_{45} := [-703010811569942788096360622140741457181943574303, \\ 2359133950371788647643796063817871044709921741609] \\ [206250160480703419278859534710705813075218221644, \\ -692125565996715873756796020149270204171356936499]$$



$$N_{46} := [-460361491896333103259524859337563623822160735323, \\ -1569158081037566807843491641983672840871715455177] \\ [-137185557429914796504140271654003591735605415116, \\ -467601721325702969333399926778755436345402421271]$$

$$N_{47} := [2768358995717297587008240401005707, \\ -20508908453895828977046357109799047] \\ [824957948652919494810532026604780, \\ -6111558173419851530072842755940537]$$

$$N_{48} := \left[ \frac{-698712892014741754166505489902359377160204367}{2}, \right. \\ \left. \frac{4689422347384309201854953920835064020229849267}{4} \right] \\ \left[ -104106576312094129260890983899094761812642801, \right. \\ \left. \frac{698712892014741754166505489902361181913204689}{2} \right]$$

$$N_{49} := \left[ \frac{-397074726172421275253684843812134445}{2}, \right. \\ \left. \frac{5883318761059670223751985896578473377}{4} \right] \\ \left[ \frac{26659253089426526822952736194350493,}{2} \right. \\ \left. \frac{-395000924306510751052288425218790757}{2} \right]$$

$$N_{410} := \left[ \frac{-1854802475109324474047850088679642277698067443}{2}, \right. \\ \left. \frac{12448535408701006001695125586831847496712873647}{4} \right] \\ \left[ \frac{124529831176821449990103535085350844783642259,}{2} \right. \\ \left. \frac{-835783881921361278554480278343662618989118139}{2} \right]$$

$$N_{411} := \left[ \frac{24931382127208596145240182958788526043353382697269}{2}, \right. \\ \left. \frac{371339050574804306226384411584741296047183920489255}{4} \right] \\ \left[ \frac{1682659419507347427142502984919596138477205063419,}{2} \right. \\ \left. \frac{25062274850727205651619554854824329882242333545821}{2} \right]$$

$$N_{412} := [1401445586364089527515609909896677321561, \\ -10382367054853350133578386455386847742168] \\ [-411157797768533245209433326356258166368, \\ 3045991378782695229592798663646039438825]$$

Fifth, he calculates the entries in the column  $U_{\varphi(f_{x_5})}$ :

```
> N[51] := MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInv
> erse(b).MatrixInverse(a).MatrixInverse(a).b;
> N[52] := c.MatrixInverse(b).MatrixInverse(a).b.d.MatrixInverse(c).Matrix
> Inverse(c);
> N[53] := d.MatrixInverse(c).MatrixInverse(c).d.MatrixInverse(c).MatrixIn
> verse(c).MatrixInverse(b).a.b.MatrixInverse(c).MatrixInverse(b).a.b.Ma
> trixInverse(c);
> N[54] := MatrixInverse(b).a.d.MatrixInverse(c).MatrixInverse(c);
> N[55] := MatrixInverse(b).MatrixInverse(a).MatrixInverse(a).MatrixInvers
> e(a).MatrixInverse(a).b.c.c.MatrixInverse(d).b;
> N[56] := MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInve
> rse(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(b).d.MatrixIn
> verse(c).MatrixInverse(c).MatrixInverse(b).a.a.b;
> N[57] := MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(b).MatrixInve
> rse(a).b.c.MatrixInverse(b).MatrixInverse(a).b.c.MatrixInverse(b).Matr
> ixInverse(a).b;
> N[58] := MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInve
> rse(b).a.b.MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).Matrix
> Inverse(c).MatrixInverse(b).MatrixInverse(a).b.c.c.MatrixInverse(d).b;
> N[59] := c.MatrixInverse(b).MatrixInverse(a).b.c.MatrixInverse(b).Matrix
> Inverse(a).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(b).Matrix
> Inverse(a).b;
> N[510] := c.MatrixInverse(b).MatrixInverse(a).b.c.MatrixInverse(b).Matri
> xInverse(a).b.d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(b).Matr
> ixInverse(a).b.c.c.MatrixInverse(d).b;
> N[511] := MatrixInverse(b).a.b.d.MatrixInverse(c).MatrixInverse(c).d.Mat
> rixInverse(c).MatrixInverse(c).d.MatrixInverse(c).MatrixInverse(c).Mat
> rixInverse(b).a.b.MatrixInverse(c);
> N[512] := MatrixInverse(b).MatrixInverse(a).MatrixInverse(a).b.d.MatrixI
> nverse(c).MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).MatrixI
> nverse(c).MatrixInverse(b);
```

$$N_{51} := \begin{bmatrix} \frac{-365385955853067554347}{2} & \frac{2452293476161570042335}{4} \\ -54441561369235201709 & \frac{365385378499638352013}{2} \end{bmatrix}$$

$$N_{52} := \begin{bmatrix} \frac{46475888407425825}{2} & \frac{692232489736400389}{4} \\ -3120351373297111 & \frac{-46475896943687759}{2} \end{bmatrix}$$

$$\begin{aligned}
 N_{53} &:= \left[ \frac{14191978774688406127415200745164963719}{2}, \right. \\
 &\quad \left. \frac{211381585887241188067953006827362787131}{4} \right] \\
 &\quad \left[ \frac{2066934845786228511332260345459789991}{2}, \right. \\
 &\quad \left. \frac{30785838434815863847055570274170645175}{2} \right] \\
 N_{54} &:= \begin{bmatrix} \frac{-373093851727}{2} & \frac{-5557025260549}{4} \\ -55590105705 & \frac{-827983683487}{2} \end{bmatrix} \\
 N_{55} &:= \begin{bmatrix} \frac{-5173498604167616695190467}{2} & \frac{34722070510940799420862545}{4} \\ -770837595439492555130851 & \frac{5173496581167136640775173}{2} \end{bmatrix} \\
 N_{56} &:= \left[ \frac{236357721809547248875036643357091258605753}{2}, \right. \\
 &\quad \left. \frac{-1586321620922972791166006152873732335817091}{4} \right] \\
 &\quad \left[ \frac{35216688575084594110611133775250811086945}{2}, \right. \\
 &\quad \left. \frac{-236357814232880135335252482498712414452247}{2} \right] \\
 N_{57} &:= [23455501903958352847442570254147219707, \\
 &\quad -78710981319565390255952867319014211859] \\
 &\quad [6989618105979490990361435765310986524, \\
 &\quad -23455464838200828841741347834673695945] \\
 N_{58} &:= [-284244077477374602753409971553960937849434307980785, \\
 &\quad 953855761412220171261967322875122763319442511726099] \\
 &\quad [-84703263165642653454673803941736597543139159266364, \\
 &\quad 284244077477374602753409971553960937985809664780451] \\
 N_{59} &:= [-137009547192400415099556348312427759, \\
 &\quad 459770844120935649284742595027142669] \\
 &\quad [18397407489614077383094633659063868, \\
 &\quad -61737241998606250592052460430053827] \\
 N_{510} &:= [-32871793295402748701492250323594559338626411841, \\
 &\quad 110309948059576753437092104294389254500749326264] \\
 &\quad [4413968139903703503835378944511234293490078624, \\
 &\quad -14812231017451052734153541914032788681367313857]
 \end{aligned}$$

$$N_{511} := \left[ \begin{array}{c} \frac{-1117059369669026324897897152421296554195}{2}, \\ \frac{-16637974509373875748385275531166037883631}{4} \end{array} \right]$$

$$\left[ \begin{array}{c} -166439216821158943893841096912313013091, \\ \frac{-2479019040546684774320083349402815695915}{2} \end{array} \right]$$

$$N_{512} := \left[ \begin{array}{cc} \frac{48347918051060432198595097427}{2} & \frac{1105492914035182258613069251201}{4} \\ 7203711790887966001556351021 & \frac{164715517453469113660801869675}{2} \end{array} \right]$$

Sixth, he calculates the entries in the column  $U_{\varphi(f_{x_6})}$ :

```
> N[61]:=MatrixInverse(c).MatrixInverse(b).MatrixInverse(a).MatrixInverse(c).MatrixInverse(b).MatrixInverse(a).MatrixInverse(c).MatrixInverse(b);
> N[62]:=c.MatrixInverse(a).d.MatrixInverse(c).MatrixInverse(b);
> N[63]:=d.MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(a).MatrixInverse(c).a.MatrixInverse(c);
> N[64]:=b.c.a.MatrixInverse(c).MatrixInverse(b);
> N[65]:=MatrixInverse(a).MatrixInverse(c).MatrixInverse(b).MatrixInverse(a).MatrixInverse(c).MatrixInverse(b).MatrixInverse(a).MatrixInverse(c).MatrixInverse(b).MatrixInverse(a);
> N[66]:=MatrixInverse(c).MatrixInverse(b).MatrixInverse(c).MatrixInverse(a).b.c.a;
> N[67]:=MatrixInverse(c).MatrixInverse(b).c.MatrixInverse(a).c.MatrixInverse(a).c.MatrixInverse(a);
> N[68]:=MatrixInverse(c).MatrixInverse(b).a.MatrixInverse(c).MatrixInverse(c).MatrixInverse(b).MatrixInverse(a);
> N[69]:=c.MatrixInverse(a).c.MatrixInverse(a).MatrixInverse(c).MatrixInverse(b).MatrixInverse(a).MatrixInverse(c).MatrixInverse(b);
> N[610]:=c.MatrixInverse(a).c.MatrixInverse(a).d.MatrixInverse(c).MatrixInverse(b).MatrixInverse(a);
> N[611]:=b.c.a.d.MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(b).a.MatrixInverse(c);
> ;
> N[612]:=MatrixInverse(a).MatrixInverse(c).MatrixInverse(b).MatrixInverse(a).MatrixInverse(c).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(b).MatrixInverse(c).MatrixInverse(b).MatrixInverse(d);
```

$$N_{61} := \left[ \begin{array}{cc} \frac{4989102626594102575547043}{2} & \frac{114077665045869559745000465}{4} \\ 334964027943461496434749 & \frac{7659075597776491896904123}{2} \end{array} \right]$$

$$N_{62} := \left[ \begin{array}{cc} \frac{5065502573}{2} & \frac{115824577897}{4} \\ -340102379 & \frac{-7776565883}{2} \end{array} \right]$$

$$\begin{aligned}
 N_{63} &:= \begin{bmatrix} \frac{7068388757136746119}{2} & \frac{105276700756844776107}{4} \\ 1029447732362045239 & \frac{15332611800004139383}{2} \end{bmatrix} \\
 N_{64} &:= \begin{bmatrix} \frac{17372872650923}{2} & \frac{397237117666257}{4} \\ -759789784787 & \frac{-17372872650685}{2} \end{bmatrix} \\
 N_{65} &:= \begin{bmatrix} \frac{122839481176503653049604626763}{2} & \frac{837434156315872177661635803007}{4} \\ 8290632772451966313990403091 & \frac{56519768682082652618974237107}{2} \end{bmatrix} \\
 N_{66} &:= \begin{bmatrix} \frac{-3695267028822070304930101}{2} & \frac{54751512591106411129906743}{4} \\ -248097026848674149289061 & \frac{3675969120329385561726419}{2} \end{bmatrix} \\
 N_{67} &:= \begin{bmatrix} -603536215070245 & -2188325322808444 \\ -81041797188976 & -293844532550381 \end{bmatrix} \\
 N_{68} &:= \begin{bmatrix} 4433429876425551979 & 15112020877128908548 \\ 595313281830109424 & 2029215978194757507 \end{bmatrix} \\
 N_{69} &:= \begin{bmatrix} 2644486227573318277141 & 30233574719374844059484 \\ -355107190573592881776 & -4059828206950396405763 \end{bmatrix} \\
 N_{610} &:= \begin{bmatrix} 302888317565353 & 1032440955663986 \\ -40672482384904 & -138638350003831 \end{bmatrix} \\
 N_{611} &:= \begin{bmatrix} \frac{-37154085868492177463035768197599}{2} & \frac{-553374013794643763898030444104547}{4} \\ 1624906569753714749910956723073 & \frac{24201404758781402065719318991873}{2} \end{bmatrix} \\
 N_{612} &:= \begin{bmatrix} \frac{-1762889723238284598642610609859449}{2}, \\ \frac{13233467868651718699587658429619917}{4} \\ \left[ -118980242945650767672293015299471, \right. \\ \left. \frac{893147881725339985446477908120247}{2} \right] \end{bmatrix}
 \end{aligned}$$

Seventh, he calculates the entries in the column  $U_{\varphi(f_{x_7})}$ :

```

> N[71]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInver
> se(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c).MatrixInver
> se(a).b.a.a.b.a.a.a;
> N[72]:=MatrixInverse(c).MatrixInverse(a).b.a.a.a.d.MatrixInverse(c).Ma
> trixInverse(c).MatrixInverse(c);
> N[73]:=d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c).d.MatrixIn
> verse(c).MatrixInverse(c).MatrixInverse(c).MatrixInverse(a).MatrixInve
> rse(a).MatrixInverse(a).MatrixInverse(b).a.c.MatrixInverse(a).MatrixIn
> verse(a).MatrixInverse(a).MatrixInverse(b).a.c;
> N[74]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInvers
> e(b).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b).d.MatrixInvers
> e(c).MatrixInverse(c).MatrixInverse(c);
> N[75]:=MatrixInverse(a).b.a.a.b.a.a.b.a.a.b.a.a.a.c.c.c.MatrixInverse(
> d).b.a.a.a;
> N[76]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInvers
> e(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c).MatrixInvers
> e(a).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b).d.MatrixInvers
> e(c).MatrixInverse(c).MatrixInverse(c).MatrixInverse(a).MatrixInverse(
> a).MatrixInverse(a).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(
> c).MatrixInverse(c).MatrixInverse(a).MatrixInverse(a).MatrixInverse(a)
> .MatrixInverse(b).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b).a
> ;
> N[77]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInvers
> e(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c).MatrixInvers
> e(c).MatrixInverse(a).b.a.a.a.MatrixInverse(c).MatrixInverse(a).b.a.a.
> a.MatrixInverse(c).MatrixInverse(a).b.a.a.a;
> N[78]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInvers
> e(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c).MatrixInvers
> e(a).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b).a.c.MatrixInve
> rse(a).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b).d.MatrixInve
> rse(c).MatrixInverse(c).MatrixInverse(c).MatrixInverse(a).b.a.a.a.c.c.
> c.MatrixInverse(d).b.a.a.a;
> N[79]:=MatrixInverse(c).MatrixInverse(a).b.a.a.a.MatrixInverse(c).Matr
> ixInverse(a).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c).Matr
> ixInverse(a).b.a.a.a;
> N[710]:=MatrixInverse(c).MatrixInverse(a).b.a.a.a.MatrixInverse(c).Matr
> ixInverse(a).b.a.a.a.d.MatrixInverse(c).MatrixInverse(c).MatrixInvers
> e(c).MatrixInverse(a).b.a.a.a.c.c.c.MatrixInverse(d).b.a.a.a;
> N[711]:=MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInver
> se(b).a.d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(c).d.MatrixI
> nverse(c).MatrixInverse(c).MatrixInverse(c).d.MatrixInverse(c).MatrixI
> nverse(c).MatrixInverse(c).MatrixInverse(a).MatrixInverse(a).MatrixInv
> erse(a).MatrixInverse(b).a.c;
> N[712]:=MatrixInverse(a).b.a.a.b.a.a.a.d.MatrixInverse(c).MatrixInvers
> e(c).MatrixInverse(c).MatrixInverse(a).MatrixInverse(a).MatrixInverse(
> a).MatrixInverse(b).d.MatrixInverse(c).MatrixInverse(c).MatrixInverse(
> c).MatrixInverse(a).MatrixInverse(a).MatrixInverse(a).MatrixInverse(b)
> ;

```

$$\begin{aligned}
 N_{71} &:= \left[ \frac{491242464152854397073164953229902461255}{2}, \right. \\
 &\quad \left. \frac{-7278569536174366155700087157823127215211}{4} \right] \\
 &\quad \left[ \frac{33154750722324811892562685660155475849}{2}, \right. \\
 &\quad \left. \frac{-491242464152854397151552695389537909577}{2} \right] \\
 N_{72} &:= \left[ \frac{25006979973079625087006197}{2} \quad \frac{372465048022695423900531297}{4} \right. \\
 &\quad \left. \frac{1678946926167764261161005}{2} \quad \frac{25006979977423521761358637}{2} \right] \\
 N_{73} &:= \left[ \frac{-7699943409566554448091575396593779468936425994969}{2}, \right. \\
 &\quad \left. \frac{114686371341339826250501196559373813819693490898411}{4} \right] \\
 &\quad \left[ \frac{-1121427927457178242018886338993276426468994958505}{2}, \right. \\
 &\quad \left. \frac{16703044799149102429509542697601018313095169457079}{2} \right] \\
 N_{74} &:= \left[ \frac{-3418963163764785449276501363}{2} \quad \frac{-50923553357916815212095363641}{4} \right. \\
 &\quad \left. \frac{-230751369629481141540301125}{2} \quad \frac{-3436913216344813651054341083}{2} \right] \\
 N_{75} &:= \left[ \frac{147884375041475017324443571313129988363901091144866247}{2}, \right. \\
 &\quad \left. \frac{-2191151591321180393736879249784838134096931438144099461}{4} \right] \\
 &\quad \left[ \frac{9978539921228100178514822142659528920999132179433159}{2}, \right. \\
 &\quad \left. \frac{-147848571705623773221439805363957597092222338109442985}{2} \right] \\
 N_{76} &:= [-1917501648134529281699752849700279323102272171442408401035 \setminus \\
 &\quad 8490170929265243493773/2, 28417838192150384328232498893822592 \setminus \\
 &\quad 5923325290235808842518059046225929258079112255/4] \\
 &\quad [-1294152965036947719482380310549512158807997829265603099818 \setminus \\
 &\quad 364022178373234676845, 1917965994558113049526115314977773565 \setminus \\
 &\quad 2018712530486691848830812359607849427713227/2] \\
 N_{77} &:= [307253407345214027463240195753018673581626320827081618934875, \\
 &\quad -2276233686806016217859636459533986431515024329647154116427101] \\
 &\quad [41474061680244923186701454603086852931821394993611576754276, \\
 &\quad -307253407345214027265109441943442824001481781885148062246249]
 \end{aligned}$$

$$N_{78} := [-1111902685260715112592331976486866225359364143481015885371 \backslash \\ 413931515528713100235212457, 8237338588069300728047160356009 \backslash \\ 289955424493157153617180340243358615333153537215506842] \\ [-1500882315655503644780069838809218869613886795424721912011 \backslash \\ 45403304812210312480745480, 11119026852607151125923319764868 \backslash \\ 66225359364143481015885371413931515491864639588207287]$$

$$N_{79} := [-7075024575670677730711671937846485302921, \\ 52414094975447925368413712383286085372946] \\ [-950022015986828434135602818386401309992, \\ 7038073669161757445953617485519799708311]$$

$$N_{710} := [-7141250992042446820444656710270462104805833586810197879182 \backslash \\ 42382304493, 52904721918219274278781127608447616944126456225 \backslash \\ 08690469944455927367053] \\ [-9589147841914006407148603823213446535969383813539132826804 \backslash \\ 3601255684, 710395420318448295524538695244804791684685771008 \backslash \\ 576162902211439464607]$$

$$N_{711} := \left[ \frac{-28219998803938529243649025390292501154743396866090613397997}{2}, \right. \\ \left. \frac{420321175095862978332646548782095579195079743209460420281847}{4} \right] \\ \left[ \frac{-1904613493343600144316411755228338922185120521766127577237,}{2}, \right. \\ \left. \frac{28368157886452135285330454729011120180899505185688266657755}{2} \right]$$

$$N_{712} := \left[ \frac{-25957886623406102864369630935798238817614699538481142186981}{2}, \right. \\ \left. \frac{-593536351831582281237697678129309863060584592765077847655111}{4} \right] \\ \left[ \frac{-1751515722129038609254521229993323948023016994642313085483,}{2}, \right. \\ \left. \frac{-40049032764891467959230675434565276507471913122150128089757}{2} \right]$$



Eight, he calculates the entries in the column  $U_{\varphi(f_{x_8})}$ :

```

> N[81]:=MatrixInverse(b).MatrixInverse(d).MatrixInverse(a);
> N[82]:=MatrixInverse(d).MatrixInverse(d).c.MatrixInverse(d).MatrixInve
> rse(b).a.d;
> N[83]:=MatrixInverse(d).MatrixInverse(b).a.MatrixInverse(b).a.d.Matrix
> Inverse(c).d.d.MatrixInverse(c).d.d;
> N[84]:=a.d.MatrixInverse(b).a.d;
> N[85]:=MatrixInverse(d).MatrixInverse(a).MatrixInverse(d).MatrixInvers
> e(a).MatrixInverse(d).MatrixInverse(a).MatrixInverse(d).MatrixInverse(
> a).MatrixInverse(d).MatrixInverse(a).b;
> N[86]:=MatrixInverse(b).a.d.MatrixInverse(b).a.d.MatrixInverse(b).a.d.
> a.d.a.d;
> N[87]:=MatrixInverse(b).a.MatrixInverse(d).c.MatrixInverse(d).MatrixIn
> verse(d).c.MatrixInverse(d).MatrixInverse(d).c;
> N[88]:=MatrixInverse(b).a.d.MatrixInverse(c).d.d.MatrixInverse(b).Matr
> ixInverse(d).MatrixInverse(a).b;
> N[89]:=MatrixInverse(d).MatrixInverse(d).c.MatrixInverse(d).MatrixInve
> rse(d).c.MatrixInverse(b);
> N[810]:=MatrixInverse(d).MatrixInverse(d).c.MatrixInverse(d).MatrixInv
> erse(d).c.MatrixInverse(d).MatrixInverse(b).MatrixInverse(d).MatrixInv
> erse(a).b;
> N[811]:=a.MatrixInverse(b).a.MatrixInverse(b).a.MatrixInverse(b).a.d.M
> atrixInverse(c).d.d;
> N[812]:=MatrixInverse(d).MatrixInverse(a).MatrixInverse(d).MatrixInver
> se(a).MatrixInverse(d).MatrixInverse(b).a.d.MatrixInverse(b).a.d.d;

```

$$N_{81} := \begin{bmatrix} \frac{2097601}{2} & \frac{14292341}{4} \\ 312537 & \frac{2129521}{2} \end{bmatrix}$$

$$N_{82} := \begin{bmatrix} \frac{-26813876357}{2} & \frac{387885462929}{4} \\ -1853791571 & \frac{26816667315}{2} \end{bmatrix}$$

$$N_{83} := \begin{bmatrix} \frac{275294109706861162119}{2} & \frac{-3981948313145580257957}{4} \\ 18579531779042231303 & \frac{-268740712634099781193}{2} \end{bmatrix}$$

$$N_{84} := \begin{bmatrix} \frac{-571570823}{2} & \frac{8268256717}{4} \\ 83886209 & \frac{-1213485159}{2} \end{bmatrix}$$

$$N_{85} := \begin{bmatrix} \frac{16738764316333488977}{2} & \frac{-112342474856279558485}{4} \\ 1157124261881365959 & \frac{-7766057328924309343}{2} \end{bmatrix}$$

$$N_{86} := \begin{bmatrix} \frac{-718483209332862630528527}{2} & \frac{10393456867555252407522837}{4} \\ -107052306989130851251047 & \frac{1548600608630707637395105}{2} \end{bmatrix}$$

$$\begin{aligned}
 N_{87} &:= \begin{bmatrix} \frac{-362905014339202517}{2} & \frac{5405268075257403809}{4} \\ -54071991727653267 & \frac{805372202374372147}{2} \end{bmatrix} \\
 N_{88} &:= \begin{bmatrix} -22095669332318830605 & 74147712742895201839 \\ -6584405441798070604 & 22095669334039730431 \end{bmatrix} \\
 N_{89} &:= \begin{bmatrix} -33830252063749 & -386771627007598 \\ -4677744857864 & -53479323358077 \end{bmatrix} \\
 N_{810} &:= \begin{bmatrix} \frac{-2530644964961716069}{2} & \frac{16984462710235839663}{4} \\ -174957482575393773 & \frac{1174229842519006355}{2} \end{bmatrix} \\
 N_{811} &:= \begin{bmatrix} \frac{2739747352948144349387}{2} & \frac{-39628644296581967709615}{4} \\ -402070084312200114547 & \frac{5815679440792026855107}{2} \end{bmatrix} \\
 N_{812} &:= \begin{bmatrix} 33187983477846157011 & -240017332068410620973 \\ 4588465451737171748 & -33184035925207219073 \end{bmatrix}
 \end{aligned}$$

Now, he is able to encrypt the ciphertext  $C'$  as the message  $S = \text{ILIKEBOB}$ .

## C.7. Example of a message, where inverse automorphisms were used for decryption in a cryptosystem based on $Aut(F)$

Bob wants to send a message to Alice. As in Example 7.0.7 let  $F$  be a free group with free generating set  $X = \{a, b, c, d\}$ , let  $\tilde{A} := \{a_1, a_2, \dots, a_{12}\} = \{A, E, I, O, U, T, M, L, K, Y, B, S\}$  be the plaintext alphabet and let  $h$  with

$$h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}} \\ x \mapsto \overline{5x + 3}$$

be the linear congruence generator. The starting seed is  $x_9 = h(x_8) = h(\overline{7324218}) = \overline{36621093}$  (the next automorphism of Example 7.0.7, which Alice and Bob use now as starting seed).

Let also

$$\tilde{U} = \{u_1, u_2, \dots, u_{12}\} \\ = \{ba^2, cd, d^2c^{-2}, a^{-1}b, a^4b^{-1}, b^3a^{-2}, bc^3, bc^{-1}bab^{-1}, c^2ba, c^2dab^{-1}, a^{-1}d^3c^{-1}, a^2db^2d^{-1}\}$$

be the free generating set for the free subgroup  $F_{\tilde{U}}$  of  $F$ .

It is known, that  $a_i \mapsto u_i$ ,  $i = 1, 2, \dots, 12$ , for  $u_i \in \tilde{U}$  and  $a_i \in \tilde{A}$ .

In GAP they define:

```
LoadPackage("FGA");;
F:=FreeGroup("a", "b", "c", "d");;
AssignGeneratorVariables(F);;
FU:=Group(b*a^2, c*d, d^2*c^-2, a^-1*b, a^4*b^-1, b^3*a^-2, b*c^3,
b*c^-1*b*a*b^-1, c^2*b*a, c^2*d*a*b^-1, a^-1*d^3*c^-1,
a^2*d*b^2*d^-1);;
```

Bob's message for Alice is

$$S = \text{YES.}$$

He first determines, with the help of the linear congruence generator  $h$ , the three automorphisms  $f_{x_9}$ ,  $f_{x_{10}}$  and  $f_{x_{11}}$ . It is

$$x_9 = h(x_8) = \overline{36621093}, \quad x_{10} = h(x_9) = \overline{183105468}, \quad x_{11} = h(x_{10}) = \overline{915527343}.$$

These automorphisms are describable with regular Nielsen transformations:

- Automorphism  $f_{x_9}$ :

$$(a, b, c, d) \xrightarrow{(N1)_3} (a, b, c^{-1}, d) \\ \xrightarrow{(N2)_{1,3}} (ac^{-1}, b, c^{-1}, d) \\ \xrightarrow{(N2)_{2,3}} (ac^{-1}, bc^{-1}, c^{-1}, d) \\ \xrightarrow{(N1)_1} (ca^{-1}, bc^{-1}, c^{-1}, d) \\ \xrightarrow{(N2)_{4,1}} (ca^{-1}, bc^{-1}, c^{-1}, dca^{-1}) \\ \xrightarrow{(N2)_{1,4}} (ca^{-1}dca^{-1}, bc^{-1}, c^{-1}, dca^{-1}) \\ \xrightarrow{(N2)_{3,4}} (ca^{-1}dca^{-1}, bc^{-1}, c^{-1}dca^{-1}, dca^{-1}) \\ \xrightarrow{(N2)_{2,3}} (ca^{-1}dca^{-1}, bc^{-2}dca^{-1}, c^{-1}dca^{-1}, dca^{-1})$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_9} : F &\rightarrow F \\
 a &\mapsto ca^{-1}dca^{-1}, \\
 b &\mapsto bc^{-2}dca^{-1}, \\
 c &\mapsto c^{-1}dca^{-1}, \\
 d &\mapsto dca^{-1}
 \end{aligned}$$

and he defines in GAP:

```

a9:=c*a^-1*d*c*a^-1;;
b9:=b*c^-2*d*c*a^-1;;
c9:=c^-1*d*c*a^-1;;
d9:=d*c*a^-1;;

```

- Automorphism  $f_{x_{10}}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N2)_{3,2}} (a, b, cb, d) \\
 &\xrightarrow{[(N2)_{1,3}]^2} (a(cb)^2, b, cb, d) \\
 &\xrightarrow{(N1)_4} (a(cb)^2, b, cb, d^{-1}) \\
 &\xrightarrow{(N2)_{3,4}} (a(cb)^2, b, cbd^{-1}, d^{-1}) \\
 &\xrightarrow{(N2)_{2,3}} (a(cb)^2, bcbd^{-1}, cbd^{-1}, d^{-1}) \\
 &\xrightarrow{(N1)_1} ((a(cb)^2)^{-1}, bcbd^{-1}, cbd^{-1}, d^{-1}) \\
 &\xrightarrow{(N4)_{4,1}} ((a(cb)^2)^{-1}, bcbd^{-1}, cbd^{-1}, d^{-1}(a(cb)^2)^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_{10}} : F &\rightarrow F \\
 a &\mapsto (a(cb)^2)^{-1}, \\
 b &\mapsto bcbd^{-1}, \\
 c &\mapsto cbd^{-1}, \\
 d &\mapsto d^{-1}(a(cb)^2)^{-1}
 \end{aligned}$$

and he defines in GAP:

```

a10:=(a*(c*b)^2)^-1;;
b10:=b*c*b*d^-1;;
c10:=c*b*d^-1;;
d10:=d^-1*(a*(c*b)^2)^-1;;

```

- Automorphism  $f_{x_{11}}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N2)_{4,1}} (a, b, c, da) \\
 &\xrightarrow{(N2)_{3,2}} (a, b, cb, da) \\
 &\xrightarrow{(N1)_4} (a, b, cb, a^{-1}d^{-1}) \\
 &\xrightarrow{(N2)_{2,4}} (a, ba^{-1}d^{-1}, cb, a^{-1}d^{-1}) \\
 &\xrightarrow{(N1)_1} (a^{-1}, ba^{-1}d^{-1}, cb, a^{-1}d^{-1}) \\
 &\xrightarrow{(N2)_{1,4}} (a^{-2}d^{-1}, ba^{-1}d^{-1}, cb, a^{-1}d^{-1}) \\
 &\xrightarrow{(N2)_{3,1}} (a^{-2}d^{-1}, ba^{-1}d^{-1}, cba^{-2}d^{-1}, a^{-1}d^{-1}) \\
 &\xrightarrow{(N2)_{4,1}} (a^{-2}d^{-1}, ba^{-1}d^{-1}, cba^{-2}d^{-1}, a^{-1}d^{-1}a^{-2}d^{-1}) \\
 &\xrightarrow{(N2)_{1,2}} (a^{-2}d^{-1}ba^{-1}d^{-1}, ba^{-1}d^{-1}, cba^{-2}d^{-1}, a^{-1}d^{-1}a^{-2}d^{-1}) \\
 &\xrightarrow{(N1)_2} (a^{-2}d^{-1}ba^{-1}d^{-1}, (ba^{-1}d^{-1})^{-1}, cba^{-2}d^{-1}, a^{-1}d^{-1}a^{-2}d^{-1}) \\
 &\xrightarrow{(N2)_{2,3}} (a^{-2}d^{-1}ba^{-1}d^{-1}, (ba^{-1}d^{-1})^{-1}cba^{-2}d^{-1}, cba^{-2}d^{-1}, a^{-1}d^{-1}a^{-2}d^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_{11}} : F &\rightarrow F \\
 a &\mapsto a^{-2}d^{-1}ba^{-1}d^{-1}, \\
 b &\mapsto (ba^{-1}d^{-1})^{-1}cba^{-2}d^{-1}, \\
 c &\mapsto cba^{-2}d^{-1}, \\
 d &\mapsto a^{-1}d^{-1}a^{-2}d^{-1}
 \end{aligned}$$

and he defines in GAP:

```

a11:=a^-2*d^-1*b*a^-1*d^-1;;
b11:=(b*a^-1*d^-1)^-1*c*b*a^-2*d^-1;;
c11:=c*b*a^-2*d^-1;;
d11:=a^-1*d^-1*a^-2*d^-1;;

```

The ciphertext from Bob is

$$\begin{aligned}
 C &= f_{x_9}(Y)f_{x_{10}}(E)f_{x_{11}}(S) \\
 &= f_{x_9}(c^2dab^{-1})f_{x_{10}}(cd)f_{x_{11}}(a^2db^2d^{-1}) \\
 &= C_1C_2C_3,
 \end{aligned}$$

with GAP he gets

```

C1:=c9^2*d9*a9*b9^-1;;
C2:=c10*d10;;
C3:=a11^2*d11*b11^2*d11^-1;;

gap> C1;
(c^-1*d*c*a^-1)^2*d*(c*a^-1)^2*c^2*b^-1

```

```

gap> C2;
      c*b*d^-2*(b^-1*c^-1)^2*a^-1
gap> C3;
      (a^-2*d^-1*b*a^-1*d^-1)^2*a^-1*d^-1*(a^-1*b^-1*c*b)^2*d*a

```

and hence the ciphertext is

$$\begin{aligned}
 C &= C_1 C_2 C_3 \\
 &= (c^{-1} d c a^{-1})^2 d (c a^{-1})^2 c^2 b^{-1} \wr \\
 &\quad c b d^{-2} (b^{-1} c^{-1})^2 a^{-1} \wr \\
 &\quad (a^{-2} d^{-1} b a^{-1} d^{-1})^2 a^{-1} d^{-1} (a^{-1} b^{-1} c b)^2 d a.
 \end{aligned}$$

Alice uses for decryption the inverse automorphisms of  $f_{x_9}$ ,  $f_{x_{10}}$  and  $f_{x_{11}}$ , which are describable with regular Nielsen transformations as follows:

- Inverse automorphism of  $f_{x_9}$  is  $f_{x_9}^{-1}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N1)_4} (a, b, c, d^{-1}) \\
 &\xrightarrow{(N2)_{1,4}} (a d^{-1}, b, c, d^{-1}) \\
 &\xrightarrow{(N2)_{2,4}} (a d^{-1}, b d^{-1}, c, d^{-1}) \\
 &\xrightarrow{(N2)_{3,4}} (a d^{-1}, b d^{-1}, c d^{-1}, d^{-1}) \\
 &\xrightarrow{(N1)_4(N1)} (d a^{-1}, b d^{-1}, c d^{-1}, d) \\
 &\xrightarrow{(N2)_{4,1}} (d a^{-1}, b d^{-1}, c d^{-1}, d^2 a^{-1}) \\
 &\xrightarrow{(N1)_3} (d a^{-1}, b d^{-1}, d c^{-1}, d^2 a^{-1}) \\
 &\xrightarrow{(N2)_{1,3}} (d a^{-1} d c^{-1}, b d^{-1}, d c^{-1}, d^2 a^{-1}) \\
 &\xrightarrow{[(N2)_{2,3}]^2} (d a^{-1} d c^{-1}, b c^{-1} d c^{-1}, d c^{-1}, d^2 a^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_9}^{-1} : F &\rightarrow F \\
 a &\mapsto d a^{-1} d c^{-1}, \\
 b &\mapsto b c^{-1} d c^{-1}, \\
 c &\mapsto d c^{-1}, \\
 d &\mapsto d^2 a^{-1}
 \end{aligned}$$

and she defines in GAP:

```

a9i:=d*a^-1*d*c^-1;;
b9i:=b*c^-1*d*c^-1;;
c9i:=d*c^-1;;
d9i:=d^2*a^-1;;

```

- Inverse automorphism of  $f_{x_{10}}$  is  $f_{x_{10}}^{-1}$ :

$$\begin{aligned}
 (a, b, c, d) &\xrightarrow{(N1)_1} (a^{-1}, b, c, d) \\
 &\xrightarrow{(N2)_{4.1}} (a^{-1}, b, c, da^{-1}) \\
 &\xrightarrow{(N1)_3} (a^{-1}, b, c^{-1}, da^{-1}) \\
 &\xrightarrow{(N2)_{2.3}} (a^{-1}, bc^{-1}, c^{-1}, da^{-1}) \\
 &\xrightarrow{(N1)_3(N1)_4} (a^{-1}, bc^{-1}, c, ad^{-1}) \\
 &\xrightarrow{(N2)_{3.4}} (a^{-1}, bc^{-1}, cad^{-1}, ad^{-1}) \\
 &\xrightarrow{(N1)_3} (a^{-1}, bc^{-1}, (cad^{-1})^{-1}, ad^{-1}) \\
 &\xrightarrow{[(N2)_{1.3}]^2} (a^{-1}(cad^{-1})^{-2}, bc^{-1}, (cad^{-1})^{-1}, ad^{-1}) \\
 &\xrightarrow{(N1)(N1)_3} (a^{-1}(cad^{-1})^{-2}, cb^{-1}, cad^{-1}, ad^{-1}) \\
 &\xrightarrow{(N2)_{3.2}} (a^{-1}(cad^{-1})^{-2}, cb^{-1}, cad^{-1}cb^{-1}, ad^{-1}) \\
 &\xrightarrow{(N1)} (a^{-1}(cad^{-1})^{-2}, bc^{-1}, cad^{-1}cb^{-1}, ad^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{x_{10}}^{-1} : F &\rightarrow F \\
 a &\mapsto a^{-1}(cad^{-1})^{-2}, \\
 b &\mapsto bc^{-1}, \\
 c &\mapsto cad^{-1}cb^{-1}, \\
 d &\mapsto ad^{-1}
 \end{aligned}$$

and she defines in GAP:

```

a10i:=a^-1*(c*a*d^-1)^-2;;
b10i:=b*c^-1;;
c10i:=c*a*d^-1*c*b^-1;;
d10i:=a*d^-1;;

```

- Inverse automorphism of  $f_{x_{11}}$  is  $f_{x_{11}}^{-1}$ :

$$\begin{aligned}
 & (a, b, c, d) \xrightarrow{(N1)} (a, b, c^{-1}, d) \\
 & \xrightarrow{(N2)_{2,3}} (a, bc^{-1}, c^{-1}, d) \\
 & \xrightarrow{(N2)_{1,2}} (abc^{-1}, bc^{-1}, c^{-1}, d) \\
 & \xrightarrow{(N1)_1} (cb^{-1}a^{-1}, bc^{-1}, c^{-1}, d) \\
 & \xrightarrow{(N2)_{4,1}} (cb^{-1}a^{-1}, bc^{-1}, c^{-1}, dcb^{-1}a^{-1}) \\
 & \xrightarrow{(N1)} (cb^{-1}a^{-1}, bc^{-1}, c, dcb^{-1}a^{-1}) \\
 & \xrightarrow{(N2)_{3,1}} (cb^{-1}a^{-1}, bc^{-1}, c^2b^{-1}a^{-1}, dcb^{-1}a^{-1}) \\
 & \xrightarrow{(N1)_4(N1)_1} (abc^{-1}, bc^{-1}, c^2b^{-1}a^{-1}, (dcb^{-1}a^{-1})^{-1}) \\
 & \xrightarrow{(N2)_{1,4}} (abc^{-1}(dcb^{-1}a^{-1})^{-1}, bc^{-1}, c^2b^{-1}a^{-1}, (dcb^{-1}a^{-1})^{-1}) \\
 & \xrightarrow{(N1)_2} (abc^{-1}(dcb^{-1}a^{-1})^{-1}, cb^{-1}, c^2b^{-1}a^{-1}, (dcb^{-1}a^{-1})^{-1}) \\
 & \xrightarrow{(N2)_{2,4}} (abc^{-1}(dcb^{-1}a^{-1})^{-1}, cb^{-1}(dcb^{-1}a^{-1})^{-1}, c^2b^{-1}a^{-1}, \\
 & \quad (dcb^{-1}a^{-1})^{-1}) \\
 & \xrightarrow{(N2)_{4,1}} (abc^{-1}(dcb^{-1}a^{-1})^{-1}, cb^{-1}(dcb^{-1}a^{-1})^{-1}, c^2b^{-1}a^{-1}, \\
 & \quad (dcb^{-1}a^{-1})^{-1}(abc^{-1}(dcb^{-1}a^{-1})^{-1})) \\
 & \xrightarrow{(N1)_2} (abc^{-1}(dcb^{-1}a^{-1})^{-1}, (cb^{-1}(dcb^{-1}a^{-1})^{-1})^{-1}, c^2b^{-1}a^{-1}, \\
 & \quad (dcb^{-1}a^{-1})^{-1}(abc^{-1}(dcb^{-1}a^{-1})^{-1})) \\
 & \xrightarrow{(N2)_{3,2}} (abc^{-1}(dcb^{-1}a^{-1})^{-1}, (cb^{-1}(dcb^{-1}a^{-1})^{-1})^{-1}, \\
 & \quad c^2b^{-1}a^{-1}(cb^{-1}(dcb^{-1}a^{-1})^{-1})^{-1}, \\
 & \quad (dcb^{-1}a^{-1})^{-1}(abc^{-1}(dcb^{-1}a^{-1})^{-1})) \\
 & \xrightarrow{(N1)_1(N1)_2} ((abc^{-1}(dcb^{-1}a^{-1})^{-1})^{-1}, cb^{-1}(dcb^{-1}a^{-1})^{-1}, \\
 & \quad c^2b^{-1}a^{-1}(cb^{-1}(dcb^{-1}a^{-1})^{-1})^{-1}, \\
 & \quad (dcb^{-1}a^{-1})^{-1}(abc^{-1}(dcb^{-1}a^{-1})^{-1}))
 \end{aligned}$$

In GAP Alice proves if she can write these elements in an equivalent shorter way:

```

a11ir:=((a*b*c^-1)*((d*c*b^-1*a^-1)^-1))^^-1;;
b11ir:=c*b^-1*(d*c*b^-1*a^-1)^^-1;;
c11ir:=c^2*b^-1*a^-1*(c*b^-1*(d*c*b^-1*a^-1)^-1)^^-1;;
d11ir:=(d*c*b^-1*a^-1)^^-1*(a*b*c^-1*(d*c*b^-1*a^-1)^^-1);;

gap> a11ir;
d*(c*b^-1*a^-1)^2
gap> b11ir;
c*b^-1*a*b*c^-1*d^-1
gap> c11ir;
c^2*b^-1*a^-1*d*c*b^-1*a^-1*b*c^-1
gap> d11ir;

```



$$a*b*c^{-1}*d^{-1}*(a*b*c^{-1})^2*d^{-1}$$

Hence, the automorphism is

$$\begin{aligned} f_{x_{11}}^{-1} : F &\rightarrow F \\ a &\mapsto d(cb^{-1}a^{-1})^2, \\ b &\mapsto cb^{-1}abc^{-1}d^{-1}, \\ c &\mapsto c^2b^{-1}a^{-1}dcb^{-1}a^{-1}bc^{-1}, \\ d &\mapsto abc^{-1}d^{-1}(abc^{-1})^2d^{-1} \end{aligned}$$

and she defines in GAP:

```
a11i:=d*(c*b^-1*a^-1)^2;;
b11i:=c*b^-1*a*b*c^-1*d^-1;;
c11i:=c^2*b^-1*a^-1*d*c*b^-1*a^-1*b*c^-1;;
d11i:=a*b*c^-1*d^-1*(a*b*c^-1)^2*d^-1;;
```

Before Alice decrypt the ciphertext she first proves if she gets the correct inverse automorphisms. Thus, she proves in GAP that  $f_{x_i}^{-1}(f_{x_i}(x_j)) = x_j$  for  $i = 9, 10, 11$  and  $j = 1, 2, 3, 4$ :

```
#Automorphism f_{x_9}
a9p:=c*a^-1*d*c*a^-1;;
b9p:=b*c^-2*d*c*a^-1;;
c9p:=c^-1*d*c*a^-1;;
d9p:=d*c*a^-1;;

#Inverse automorphism f^{-1}_{x_9}
a9ip:=d9p*a9p^-1*d9p*c9p^-1;;
b9ip:=b9p*c9p^-1*d9p*c9p^-1;;
c9ip:=d9p*c9p^-1;;
d9ip:=d9p^2*a9p^-1;;

gap> a9ip; b9ip; c9ip; d9ip;
a
b
c
d

#####

#Automorphism f_{x_10}
a10p:=(a*(c*b)^2)^-1;;
b10p:=b*c*b*d^-1;;
c10p:=c*b*d^-1;;
d10p:=d^-1*(a*(c*b)^2)^-1;;

#Inverse automorphism f^{-1}_{x_10}
a10ip:=a10p^-1*(c10p*a10p*d10p^-1)^-2;;
b10ip:=b10p*c10p^-1;;
c10ip:=c10p*a10p*d10p^-1*c10p*b10p^-1;;
d10ip:=a10p*d10p^-1;;
```

```

gap> a10ip; b10ip; c10ip; d10ip;
a
b
c
d

#####

#Automorphism f_{x_11}
a11p:=a^-2*d^-1*b*a^-1*d^-1;;
b11p:=(b*a^-1*d^-1)^-1*c*b*a^-2*d^-1;;
c11p:=c*b*a^-2*d^-1;;
d11p:=a^-1*d^-1*a^-2*d^-1;;

#Inverse automorphism f^{-1}_{x_11}
a11ip:=d11p*(c11p*b11p^-1*a11p^-1)^2;;
b11ip:=c11p*b11p^-1*a11p*b11p*c11p^-1*d11p^-1;;
c11ip:=c11p^2*b11p^-1*a11p^-1*d11p*c11p*b11p^-1*a11p^-1*b11p*c11p^-1;;
d11ip:=a11p*b11p*c11p^-1*d11p^-1*(a11p*b11p*c11p^-1)^2*d11p^-1;;

gap> a11ip; b11ip; c11ip; d11ip;
a
b
c
d

```

To decrypt the ciphertext

$$\begin{aligned}
 C &= C_1 C_2 C_3 \\
 &= (c^{-1} d c a^{-1})^2 d (c a^{-1})^2 c^2 b^{-1} \\
 &\quad c b d^{-2} (b^{-1} c^{-1})^2 a^{-1} \\
 &\quad (a^{-2} d^{-1} b a^{-1} d^{-1})^2 a^{-1} d^{-1} (a^{-1} b^{-1} c b)^2 d a,
 \end{aligned}$$

which she gets from Bob, she calculates

$$\begin{aligned}
 S &= f_{x_9}^{-1}(C_1) f_{x_{10}}^{-1}(C_2) f_{x_{11}}^{-1}(C_3) \\
 &= f_{x_9}^{-1}((c^{-1} d c a^{-1})^2 d (c a^{-1})^2 c^2 b^{-1}) f_{x_{10}}^{-1}(c b d^{-2} (b^{-1} c^{-1})^2 a^{-1}) \\
 &\quad f_{x_{11}}^{-1}((a^{-2} d^{-1} b a^{-1} d^{-1})^2 a^{-1} d^{-1} (a^{-1} b^{-1} c b)^2 d a) \\
 &= S_1 S_2 S_3,
 \end{aligned}$$

in GAP, this is:

```

S1:=(c9i^-1*d9i*c9i*a9i^-1)^2*d9i*(c9i*a9i^-1)^2*c9i^2*b9i^-1;;
S2:=c10i*b10i*d10i^-2*(b10i^-1*c10i^-1)^2*a10i^-1;;
S3:=(a11i^-2*d11i^-1*b11i*a11i^-1*d11i^-1)^2*a11i^-1*d11i^-1*\
(a11i^-1*b11i^-1*c11i*b11i)^2*d11i*a11i;;

gap> S1;

```

C.7. Example of a message, where inverse automorphisms were used for decryption in a cryptosystem based on  $\text{Aut}(F)$

---

```
      c^2*d*a*b^-1
gap> S2;
      c*d
gap> S3;
      a^2*d*b^2*d^-1
```

With the one-to-one correspondence between the plaintext alphabet  $\tilde{A}$  and the Nielsen reduced generating set  $\tilde{U}$  for the subgroup  $F_{\tilde{U}}$  Alice can read the message YES from Bob.

## C.8. Example 8.0.4 calculated with GAP

Alice and Bob use the free group  $F = \langle X \mid \ \rangle$ , with free generating set  $X = \{x, y, z\}$ , and the explicit free subgroup  $F_U$  of  $F$ , with free generating set  $U = \{u_1, u_2, \dots, u_8\}$ ,  $u_i$  words in  $X$ , they choose

$$\begin{aligned} u_1 &:= xyz, & u_2 &:= yzy^{-1}, & u_3 &:= x^{-1}zx^{-1}, & u_4 &:= y^{-1}x^2, \\ u_5 &:= z^{-1}xyx, & u_6 &:= z^{-1}yx^{-1}, & u_7 &:= x^3y, & u_8 &:= y^3z^{-2}. \end{aligned}$$

In GAP they define

```
LoadPackage("FGA");;
F:=FreeGroup("x", "y", "z");;
AssignGeneratorVariables(F);;

u1:=x*y*z;;
u2:=y*z*y^-1;;
u3:=x^-1*z*x^-1;;
u4:=y^-1*x^2;;
u5:=z^-1*x*y*x;;
u6:=z^-1*y*x^-1;;
u7:=x^3*y;;
u8:=y^3*z^-2;;

FU:=Group(u1, u2, u3, u4, u5, u6, u7, u8);;
```

and prove that  $U$  is a Nielsen reduced set with the operation

▷ `FreeGeneratorsOfGroup(FU)`

which gives a Nielsen reduced generating set for the group  $FU$ :

```
gap> FreeGeneratorsOfGroup(FU);
[ x*y*z, y*z*y^-1, x^-1*z*x^-1, y^-1*x^2, z^-1*x*y*x, \
  z^-1*y*x^-1, x^3*y, y^3*z^-2 ]
```

Alice knows the linear congruence generator  $h$  hence she can get the 4 required automorphisms of the set  $\mathcal{H}_{Aut}$  to encrypt her message.

These automorphisms are describable with Nielsen transformations as follows:

- Automorphism  $f_{u_1}$ :

$$\begin{aligned}
& (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8) \\
& \xrightarrow{(N2)_{1.7}} (u_1u_7, u_2, u_3, u_4, u_5, u_6, u_7, u_8) \\
& \xrightarrow{(N2)_{2.4}} (u_1u_7, u_2u_4, u_3, u_4, u_5, u_6, u_7, u_8) \\
& \xrightarrow{(N1)_5} (u_1u_7, u_2u_4, u_3, u_4, u_5^{-1}, u_6, u_7, u_8) \\
& \xrightarrow{(N2)_{7.8}} (u_1u_7, u_2u_4, u_3, u_4, u_5^{-1}, u_6, u_7u_8, u_8) \\
& \xrightarrow{[(N2)_{3.4}]^2} (u_1u_7, u_2u_4, u_3u_4^2, u_4, u_5^{-1}, u_6, u_7u_8, u_8) \\
& \xrightarrow{(N2)_{4.6}} (u_1u_7, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}, u_6, u_7u_8, u_8) \\
& \xrightarrow{(N2)_{5.1}} (u_1u_7, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6, u_7u_8, u_8) \\
& \xrightarrow{(N1)_7} (u_1u_7, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6, u_8^{-1}u_7^{-1}, u_8) \\
& \xrightarrow{(N2)_{6.3}} (u_1u_7, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6u_3u_4^2, u_8^{-1}u_7^{-1}, u_8) \\
& \xrightarrow{(N2)_{8.1}} (u_1u_7, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6u_3u_4^2, u_8^{-1}u_7^{-1}, u_8u_1u_7) \\
& \xrightarrow{(N2)_{7.4}} (u_1u_7, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6u_3u_4^2, u_8^{-1}u_7^{-1}u_4u_6, u_8u_1u_7) \\
& \xrightarrow{(N1)_7} (u_1u_7, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6u_3u_4^2, u_6^{-1}u_4^{-1}u_7u_8, u_8u_1u_7) \\
& \xrightarrow{(N2)_{1.2}} (u_1u_7u_2u_4, u_2u_4, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6u_3u_4^2, u_6^{-1}u_4^{-1}u_7u_8, u_8u_1u_7) \\
& \xrightarrow{(N2)_{2.3}} (u_1u_7u_2u_4, u_2u_4u_3u_4^2, u_3u_4^2, u_4u_6, u_5^{-1}u_1u_7, u_6u_3u_4^2, u_6^{-1}u_4^{-1}u_7u_8, u_8u_1u_7) \\
& \xrightarrow{(N2)_{4.5}} (u_1u_7u_2u_4, u_2u_4u_3u_4^2, u_3u_4^2, u_4u_6u_5^{-1}u_1u_7, u_5^{-1}u_1u_7, u_6u_3u_4^2, u_6^{-1}u_4^{-1}u_7u_8, u_8u_1u_7)
\end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
f_{u_1} : H &\rightarrow H \\
u_1 &\mapsto u_1u_7u_2u_4, \\
u_2 &\mapsto u_2u_4u_3u_4^2, \\
u_3 &\mapsto u_3u_4^2, \\
u_4 &\mapsto u_4u_6u_5^{-1}u_1u_7, \\
u_5 &\mapsto u_5^{-1}u_1u_7, \\
u_6 &\mapsto u_6u_3u_4^2, \\
u_7 &\mapsto u_6^{-1}u_4^{-1}u_7u_8, \\
u_8 &\mapsto u_8u_1u_7.
\end{aligned}$$

- Automorphism  $f_{u_2}$ :

$$\begin{aligned}
 & (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8) \\
 & \xrightarrow{(N2)_{1.3}} (u_1u_3, u_2, u_3, u_4, u_5, u_6, u_7, u_8) \\
 & \xrightarrow{(N2)_{3.5}} (u_1u_3, u_2, u_3u_5, u_4, u_5, u_6, u_7, u_8) \\
 & \xrightarrow{(N1)_2(N1)_4} (u_1u_3, u_2^{-1}, u_3u_5, u_4^{-1}, u_5, u_6, u_7, u_8) \\
 & \xrightarrow{(N2)_{6.5}} (u_1u_3, u_2^{-1}, u_3u_5, u_4^{-1}, u_5, u_6u_5, u_7, u_8) \\
 & \xrightarrow{(N1)_1} (u_3^{-1}u_1^{-1}, u_2^{-1}, u_3u_5, u_4^{-1}, u_5, u_6u_5, u_7, u_8) \\
 & \xrightarrow{[(N2)_{3.4}]^2} (u_3^{-1}u_1^{-1}, u_2^{-1}, u_3u_5u_4^{-2}, u_4^{-1}, u_5, u_6u_5, u_7, u_8) \\
 & \xrightarrow{(N2)_{5.2}} (u_3^{-1}u_1^{-1}, u_2^{-1}, u_3u_5u_4^{-2}, u_4^{-1}, u_5u_2^{-1}, u_6u_5, u_7, u_8) \\
 & \xrightarrow{(N2)_{7.6}} (u_3^{-1}u_1^{-1}, u_2^{-1}, u_3u_5u_4^{-2}, u_4^{-1}, u_5u_2^{-1}, u_6u_5, u_7u_6u_5, u_8) \\
 & \xrightarrow{(N2)_{4.2}} (u_3^{-1}u_1^{-1}, u_2^{-1}, u_3u_5u_4^{-2}, u_4^{-1}u_2^{-1}, u_5u_2^{-1}, u_6u_5, u_7u_6u_5, u_8) \\
 & \xrightarrow{(N2)_{2.8}} (u_3^{-1}u_1^{-1}, u_2^{-1}u_8, u_3u_5u_4^{-2}, u_4^{-1}u_2^{-1}, u_5u_2^{-1}, u_6u_5, u_7u_6u_5, u_8) \\
 & \xrightarrow{(N2)_{8.4}} (u_3^{-1}u_1^{-1}, u_2^{-1}u_8, u_3u_5u_4^{-2}, u_4^{-1}u_2^{-1}, u_5u_2^{-1}, u_6u_5, u_7u_6u_5, u_8u_4^{-1}u_2^{-1}) \\
 & \xrightarrow{(N1)_4} (u_3^{-1}u_1^{-1}, u_2^{-1}u_8, u_3u_5u_4^{-2}, u_2u_4, u_5u_2^{-1}, u_6u_5, u_7u_6u_5, u_8u_4^{-1}u_2^{-1}) \\
 & \xrightarrow{(N2)_{1.4}} (u_3^{-1}u_1^{-1}u_2u_4, u_2^{-1}u_8, u_3u_5u_4^{-2}, u_2u_4, u_5u_2^{-1}, u_6u_5, u_7u_6u_5, u_8u_4^{-1}u_2^{-1}) \\
 & \xrightarrow{(N2)_{2.6}} (u_3^{-1}u_1^{-1}u_2u_4, u_2^{-1}u_8u_6u_5, u_3u_5u_4^{-2}, u_2u_4, u_5u_2^{-1}, u_6u_5, u_7u_6u_5, u_8u_4^{-1}u_2^{-1}) \\
 & \xrightarrow{(N2)_{5.6}} (u_3^{-1}u_1^{-1}u_2u_4, u_2^{-1}u_8u_6u_5, u_3u_5u_4^{-2}, u_2u_4, u_5u_2^{-1}u_6u_5, u_6u_5, u_7u_6u_5, u_8u_4^{-1}u_2^{-1}) \\
 & \xrightarrow{(N2)_{6.4}} (u_3^{-1}u_1^{-1}u_2u_4, u_2^{-1}u_8u_6u_5, u_3u_5u_4^{-2}, u_2u_4, u_5u_2^{-1}u_6u_5, u_6u_5u_2u_4, u_7u_6u_5, u_8u_4^{-1}u_2^{-1}) \\
 & \xrightarrow{(N2)_{4.7}} (u_3^{-1}u_1^{-1}u_2u_4, u_2^{-1}u_8u_6u_5, u_3u_5u_4^{-2}, u_2u_4u_7u_6u_5, u_5u_2^{-1}u_6u_5, u_6u_5u_2u_4, u_7u_6u_5, u_8u_4^{-1}u_2^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{u_2} : H &\rightarrow H \\
 u_1 &\mapsto u_3^{-1}u_1^{-1}u_2u_4, \\
 u_2 &\mapsto u_2^{-1}u_8u_6u_5, \\
 u_3 &\mapsto u_3u_5u_4^{-2}, \\
 u_4 &\mapsto u_2u_4u_7u_6u_5, \\
 u_5 &\mapsto u_5u_2^{-1}u_6u_5, \\
 u_6 &\mapsto u_6u_5u_2u_4, \\
 u_7 &\mapsto u_7u_6u_5, \\
 u_8 &\mapsto u_8u_4^{-1}u_2^{-1}.
 \end{aligned}$$

- Automorphism  $f_{u_3}$ :

$$\begin{aligned}
& (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8) \\
& \xrightarrow{(N1)_2(N1)_5(N1)_8} (u_1, u_2^{-1}, u_3, u_4, u_5^{-1}, u_6, u_7, u_8^{-1}) \\
& \xrightarrow{(N2)_{6.3}} (u_1, u_2^{-1}, u_3, u_4, u_5^{-1}, u_6 u_3, u_7, u_8^{-1}) \\
& \xrightarrow{(N2)_{3.7}} (u_1, u_2^{-1}, u_3 u_7, u_4, u_5^{-1}, u_6 u_3, u_7, u_8^{-1}) \\
& \xrightarrow{(N2)_{1.2}} (u_1 u_2^{-1}, u_2^{-1}, u_3 u_7, u_4, u_5^{-1}, u_6 u_3, u_7, u_8^{-1}) \\
& \xrightarrow{[(N2)_{4.8}]^2} (u_1 u_2^{-1}, u_2^{-1}, u_3 u_7, u_4 u_8^{-2}, u_5^{-1}, u_6 u_3, u_7, u_8^{-1}) \\
& \xrightarrow{(N2)_{5.6}} (u_1 u_2^{-1}, u_2^{-1}, u_3 u_7, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3, u_7, u_8^{-1}) \\
& \xrightarrow{(N2)_{8.3}} (u_1 u_2^{-1}, u_2^{-1}, u_3 u_7, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3, u_7, u_8^{-1} u_3 u_7) \\
& \xrightarrow{(N2)_{6.3}} (u_1 u_2^{-1}, u_2^{-1}, u_3 u_7, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3^2 u_7, u_7, u_8^{-1} u_3 u_7) \\
& \xrightarrow{(N1)_8} (u_1 u_2^{-1}, u_2^{-1}, u_3 u_7, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3^2 u_7, u_7, u_7^{-1} u_3^{-1} u_8) \\
& \xrightarrow{(N2)_{2.3}} (u_1 u_2^{-1}, u_2^{-1} u_3 u_7, u_3 u_7, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3^2 u_7, u_7, u_7^{-1} u_3^{-1} u_8) \\
& \xrightarrow{(N2)_{7.4}} (u_1 u_2^{-1}, u_2^{-1} u_3 u_7, u_3 u_7, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3^2 u_7, u_7 u_4 u_8^{-2}, u_7^{-1} u_3^{-1} u_8) \\
& \xrightarrow{(N2)_{1.8}} (u_1 u_2^{-1} u_7^{-1} u_3^{-1} u_8, u_2^{-1} u_3 u_7, u_3 u_7, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3^2 u_7, u_7 u_4 u_8^{-2}, u_7^{-1} u_3^{-1} u_8) \\
& \xrightarrow{(N2)_{3.4}} (u_1 u_2^{-1} u_7^{-1} u_3^{-1} u_8, u_2^{-1} u_3 u_7, u_3 u_7 u_4 u_8^{-2}, u_4 u_8^{-2}, u_5^{-1} u_6 u_3, u_6 u_3^2 u_7, u_7 u_4 u_8^{-2}, u_7^{-1} u_3^{-1} u_8)
\end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
f_{u_3} : H &\rightarrow H \\
u_1 &\mapsto u_1 u_2^{-1} u_7^{-1} u_3^{-1} u_8, \\
u_2 &\mapsto u_2^{-1} u_3 u_7, \\
u_3 &\mapsto u_3 u_7 u_4 u_8^{-2}, \\
u_4 &\mapsto u_4 u_8^{-2}, \\
u_5 &\mapsto u_5^{-1} u_6 u_3, \\
u_6 &\mapsto u_6 u_3^2 u_7, \\
u_7 &\mapsto u_7 u_4 u_8^{-2}, \\
u_8 &\mapsto u_7^{-1} u_3^{-1} u_8.
\end{aligned}$$

- Automorphism  $f_{u_4}$ :

$$\begin{aligned}
 & (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8) \\
 & \xrightarrow{(N1)_1(N1)_3(N1)_4} (u_1^{-1}, u_2, u_3^{-1}, u_4^{-1}, u_5, u_6, u_7, u_8) \\
 & \xrightarrow{(N2)_{6.2}} (u_1^{-1}, u_2, u_3^{-1}, u_4^{-1}, u_5, u_6 u_2, u_7, u_8) \\
 & \xrightarrow{[(N2)_{8.2}]^3} (u_1^{-1}, u_2, u_3^{-1}, u_4^{-1}, u_5, u_6 u_2, u_7, u_8 u_2^3) \\
 & \xrightarrow{(N2)_{2.3}} (u_1^{-1}, u_2 u_3^{-1}, u_3^{-1}, u_4^{-1}, u_5, u_6 u_2, u_7, u_8 u_2^3) \\
 & \xrightarrow{(N2)_{3.4}} (u_1^{-1}, u_2 u_3^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1}, u_5, u_6 u_2, u_7, u_8 u_2^3) \\
 & \xrightarrow{(N2)_{5.2}} (u_1^{-1}, u_2 u_3^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7, u_8 u_2^3) \\
 & \xrightarrow{(N2)_{7.4}} (u_1^{-1}, u_2 u_3^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1}, u_8 u_2^3) \\
 & \xrightarrow{(N2)_{1.3}} (u_1^{-1} u_3^{-1} u_4^{-1}, u_2 u_3^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1}, u_8 u_2^3) \\
 & \xrightarrow{(N2)_{4.5}} (u_1^{-1} u_3^{-1} u_4^{-1}, u_2 u_3^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1}, u_8 u_2^3) \\
 & \xrightarrow{(N2)_{8.3}} (u_1^{-1} u_3^{-1} u_4^{-1}, u_2 u_3^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1}, u_8 u_2^3 u_3^{-1} u_4^{-1}) \\
 & \xrightarrow{(N1)_1(N1)_2} (u_4 u_3 u_1, u_3 u_2^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1}, u_8 u_2^3 u_3^{-1} u_4^{-1}) \\
 & \xrightarrow{(N2)_{7.2}} (u_4 u_3 u_1, u_3 u_2^{-1}, u_3^{-1} u_4^{-1}, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1} u_3 u_2^{-1}, u_8 u_2^3 u_3^{-1} u_4^{-1}) \\
 & \xrightarrow{(N1)_3} (u_4 u_3 u_1, u_3 u_2^{-1}, u_4 u_3, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1} u_3 u_2^{-1}, u_8 u_2^3 u_3^{-1} u_4^{-1}) \\
 & \xrightarrow{(N2)_{2.3}} (u_4 u_3 u_1, u_3 u_2^{-1} u_4 u_3, u_4 u_3, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1} u_3 u_2^{-1}, u_8 u_2^3 u_3^{-1} u_4^{-1}) \\
 & \xrightarrow{(N2)_{3.5}} (u_4 u_3 u_1, u_3 u_2^{-1} u_4 u_3, u_4 u_3 u_5 u_2 u_3^{-1}, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2, u_7 u_4^{-1} u_3 u_2^{-1}, u_8 u_2^3 u_3^{-1} u_4^{-1}) \\
 & \xrightarrow{(N2)_{6.1}} (u_4 u_3 u_1, u_3 u_2^{-1} u_4 u_3, u_4 u_3 u_5 u_2 u_3^{-1}, u_4^{-1} u_5 u_2 u_3^{-1}, u_5 u_2 u_3^{-1}, u_6 u_2 u_4 u_3 u_1, u_7 u_4^{-1} u_3 u_2^{-1}, u_8 u_2^3 u_3^{-1} u_4^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 f_{u_4} : H & \rightarrow H \\
 u_1 & \mapsto u_4 u_3 u_1, \\
 u_2 & \mapsto u_3 u_2^{-1} u_4 u_3, \\
 u_3 & \mapsto u_4 u_3 u_5 u_2 u_3^{-1}, \\
 u_4 & \mapsto u_4^{-1} u_5 u_2 u_3^{-1}, \\
 u_5 & \mapsto u_5 u_2 u_3^{-1}, \\
 u_6 & \mapsto u_6 u_2 u_4 u_3 u_1, \\
 u_7 & \mapsto u_7 u_4^{-1} u_3 u_2^{-1}, \\
 u_8 & \mapsto u_8 u_2^3 u_3^{-1} u_4^{-1}.
 \end{aligned}$$

In GAP she defines for the automorphisms:

```

#Automorphism f_{u_1}
u11:=u1*u7*u2*u4;;
u12:=u2*u4*u3*u4^2;;
u13:=u3*u4^2;;

```



```

u14:=u4*u6*u5^-1*u1*u7;;
u15:=u5^-1*u1*u7;;
u16:=u6*u3*u4^2;;
u17:=u6^-1*u4^-1*u7*u8;;
u18:=u8*u1*u7;;

#Automorphism f_{u_2}
u21:=u3^-1*u1^-1*u2*u4;;
u22:=u2^-1*u8*u6*u5;;
u23:=u3*u5*u4^-2;;
u24:=u2*u4*u7*u6*u5;;
u25:=u5*u2^-1*u6*u5;;
u26:=u6*u5*u2*u4;;
u27:=u7*u6*u5;;
u28:=u8*u4^-1*u2^-1;;

#Automorphism f_{u_3}
u31:=u1*u2^-1*u7^-1*u3^-1*u8;;
u32:=u2^-1*u3*u7;;
u33:=u3*u7*u4*u8^-2;;
u34:=u4*u8^-2;;
u35:=u5^-1*u6*u3;;
u36:=u6*u2^3*u7;;
u37:=u7*u4*u8^-2;;
u38:=u7^-1*u3^-1*u8;;

#Automorphism f_{u_4}
u41:=u4*u3*u1;;
u42:=u3*u2^-1*u4*u3;;
u43:=u4*u3*u5*u2*u3^-1;;
u44:=u4^-1*u5*u2*u3^-1;;
u45:=u5*u2*u3^-1;;
u46:=u6*u2*u4*u3*u1;;
u47:=u7*u4^-1*u3*u2^-1;;
u48:=u8*u2^3*u3^-1*u4^-1;;

```

Hence, to get the ciphertext

$$\begin{aligned}
C &= f_{u_1}(L)f_{u_2}(O)f_{u_3}(V)f_{u_4}(E) \\
&= f_{u_1}(u_1)f_{u_2}(u_4)f_{u_3}(u_7)f_{u_4}(u_2)
\end{aligned}$$

as a word in  $X$ , she calculates in GAP:

```

gap> u11;
x*y*z*x^3*y^2*z*y^-2*x^2
gap> u24;
y*z*y^-2*x^5*y*z^-1*y*x^-1*z^-1*x*y*x
gap> u37;
x^5*(z^2*y^-3)^2
gap> u42;
x^-1*z*x^-1*y*z^-1*y^-2*x*z*x^-1

```

Thus, the ciphertext is

$$C = xyzx^3y^2zy^{-2}x^2 \wr yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx \wr x^5(z^2y^{-3})^2 \wr x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1}$$

and this is sent to Bob.

For decryption Bob calculates the tables Table 8.3 (page 194) and Table 8.4 (page 194). For this he chooses the automorphisms in  $\mathcal{H}_{aut}$ , which Alice also used. In GAP it is:

```

gap> u11; u12; u13; u14; u15; u16; u17; u18;
x*y*z*x^3*y^2*z*y^-2*x^2
y*z*y^-2*x*z*x^-1*(y^-1*x^2)^2
x^-1*z*x^-1*(y^-1*x^2)^2
y^-1*x^2*z^-1*y*x^-2*y^-1*x^-1*z*x*y*z*x^3*y
x^-1*y^-1*x^-1*z*x*y*z*x^3*y
z^-1*y*x^-2*z*x^-1*(y^-1*x^2)^2
x*y^-1*z*x^-2*y*x^3*y^4*z^-2
y^3*z^-2*x*y*z*x^3*y

gap> u21; u22; u23; u24; u25; u26; u27; u28;
(x*z^-1)^2*y^-1*x^-1*y*z*y^-2*x^2
y*z^-1*y^2*z^-3*y*x^-1*z^-1*x*y*x
x^-1*z*x^-1*z^-1*x*(y*x^-1)^2*x^-1*y
y*z*y^-2*x^5*y*z^-1*y*x^-1*z^-1*x*y*x
z^-1*(x*y)^2*z^-1*y^-1*z^-1*y*x^-1*z^-1*x*y*x
z^-1*y*x^-1*z^-1*(x*y)^2*z*y^-2*x^2
x^3*y*z^-1*y*x^-1*z^-1*x*y*x
y^3*z^-2*x^-2*y^2*z^-1*y^-1

gap> u31; u32; u33; u34; u35; u36; u37; u38;
x*y*z*y*z^-1*y^-2*x^-2*z^-1*x*y^3*z^-2
y*z^-1*y^-1*x^-1*z*x^2*y
x^-1*z*x^4*(z^2*y^-3)^2
y^-1*x^2*(z^2*y^-3)^2
x^-1*y^-1*x^-1*y*x^-2*z*x^-1
z^-1*y*x^-1*y*z^3*y^-1*x^3*y
x^5*(z^2*y^-3)^2
y^-1*x^-2*z^-1*x*y^3*z^-2

gap> u41; u42; u43; u44; u45; u46; u47; u48;
y^-1*x*z*y*z
x^-1*z*x^-1*y*z^-1*y^-2*x*z*x^-1
y^-1*x*z*x^-1*z^-1*(x*y)^2*z*y^-1*x*z^-1*x
x^-2*y*z^-1*(x*y)^2*z*y^-1*x*z^-1*x
z^-1*(x*y)^2*z*y^-1*x*z^-1*x
z^-1*y*x^-1*y*z*y^-2*x*z*y*z
x^3*y*x^-2*y*x^-1*z*x^-1*y*z^-1*y^-1
y^3*z^-2*y*z^3*y^-1*x*z^-1*x^-1*y

```

With this information Bob is able to reconstruct the message  $S = \text{LOVE}$ .

## C.9. Example for decryption where Bob uses an algorithm to solve a constructive membership problem for a cryptosystem based on $Aut(F_U)$

We are in the situation of Example 8.0.4, that means Bob and Alice agreed on the following **public parameters**.

1. Let  $F$  be the free group on the free generating set  $X = \{x, y, z\}$ .
2. Let  $\tilde{A} = \{a_1, a_2, \dots, a_8\} = \{L, E, I, O, U, A, V, B\}$  be the plaintext alphabet.
3. Let  $H$  be the abstract free group of rank  $|\tilde{A}| = 8$  with free generating set  $U = \{u_1, u_2, \dots, u_8\}$ .
4. A set  $\mathcal{H}_{Aut} \subset Aut(H)$  is determined. The automorphisms, which Alice and Bob use for encryption and decryption, respectively, are just given at the moment when they are needed.
5. The linear congruence generator with maximal periodic length is

$$h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}}$$

$$u \mapsto \overline{133u + 51}.$$

The **private parameters** are the following:

Let  $F_U$  be the explicit finitely generated free group, which is generated with the free generating set  $U = \{u_1, u_2, \dots, u_8\}$  with words in  $X$ , for this example it is

$$\begin{aligned} u_1 &:= xyz, & u_2 &:= yzy^{-1}, & u_3 &:= x^{-1}zx^{-1}, & u_4 &:= y^{-1}x^2, \\ u_5 &:= z^{-1}xyx, & u_6 &:= z^{-1}yx^{-1}, & u_7 &:= x^3y, & u_8 &:= y^3z^{-2}. \end{aligned}$$

The starting automorphism  $f_{u_1}$  is  $f_{\overline{23442}}$ , hence it is  $u_1 = \bar{\alpha} = \overline{23442}$ . It is known, that  $a_i \mapsto u_i$ ,  $i = 1, 2, \dots, 12$ , for  $u_i \in U$  and  $a_i \in \tilde{A}$ , therefore

$$\begin{aligned} L &\hat{=} u_1 = xyz, & E &\hat{=} u_2 = yzy^{-1}, & I &\hat{=} u_3 = x^{-1}zx^{-1}, & O &\hat{=} u_4 = y^{-1}x^2, \\ U &\hat{=} u_5 = z^{-1}xyx, & A &\hat{=} u_6 = z^{-1}yx^{-1}, & V &\hat{=} u_7 = x^3y, & B &\hat{=} u_8 = y^3z^{-2}. \end{aligned}$$

Now, **Bob** gets the ciphertext

$$\begin{aligned} C &= xyzx^3y^2zy^{-2}x^2 \wr yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx \wr x^5(z^2y^{-3})^2 \wr x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1} \\ &= c_1c_2c_3c_4 \end{aligned}$$

from Alice.

Bob knows that Alice used 4 automorphisms to encrypt her message. With the help of the linear congruence generator  $h : \mathbb{Z}_{2^{128}} \rightarrow \mathbb{Z}_{2^{128}}$  with  $u \mapsto \overline{133u + 51}$  and the starting seed  $\bar{\alpha} = \overline{23442}$ , he is able to reconstruct these 4 automorphisms  $f_{u_i} \in \mathcal{H}_{Aut}$ ,  $1 \leq i \leq 4$ . It is

$$\begin{aligned} u_1 &= \bar{\alpha} = \overline{23442}, & u_2 &= h(u_1) = \overline{3117837}, \\ u_3 &= h(u_2) = \overline{414672372} & \text{and} & \quad u_4 = h(u_3) = \overline{55151425527}. \end{aligned}$$

The automorphisms are

$$\begin{aligned}
 f_{u_1} : H &\rightarrow H \\
 u_1 &\mapsto u_1 u_7 u_2 u_4, & u_5 &\mapsto u_5^{-1} u_1 u_7, \\
 u_2 &\mapsto u_2 u_4 u_3 u_4^2, & u_6 &\mapsto u_6 u_3 u_4^2, \\
 u_3 &\mapsto u_3 u_4^2, & u_7 &\mapsto u_6^{-1} u_4^{-1} u_7 u_8, \\
 u_4 &\mapsto u_4 u_6 u_5^{-1} u_1 u_7, & u_8 &\mapsto u_8 u_1 u_7;
 \end{aligned}$$

$$\begin{aligned}
 f_{u_2} : H &\rightarrow H \\
 u_1 &\mapsto u_3^{-1} u_1^{-1} u_2 u_4, & u_5 &\mapsto u_5 u_2^{-1} u_6 u_5, \\
 u_2 &\mapsto u_2^{-1} u_8 u_6 u_5, & u_6 &\mapsto u_6 u_5 u_2 u_4, \\
 u_3 &\mapsto u_3 u_5 u_4^{-2}, & u_7 &\mapsto u_7 u_6 u_5, \\
 u_4 &\mapsto u_2 u_4 u_7 u_6 u_5, & u_8 &\mapsto u_8 u_4^{-1} u_2^{-1};
 \end{aligned}$$

$$\begin{aligned}
 f_{u_3} : H &\rightarrow H \\
 u_1 &\mapsto u_1 u_2^{-1} u_7^{-1} u_3^{-1} u_8, & u_5 &\mapsto u_5^{-1} u_6 u_3, \\
 u_2 &\mapsto u_2^{-1} u_3 u_7, & u_6 &\mapsto u_6 u_3^2 u_7, \\
 u_3 &\mapsto u_3 u_7 u_4 u_8^{-2}, & u_7 &\mapsto u_7 u_4 u_8^{-2}, \\
 u_4 &\mapsto u_4 u_8^{-2}, & u_8 &\mapsto u_7^{-1} u_3^{-1} u_8;
 \end{aligned}$$

$$\begin{aligned}
 f_{u_4} : H &\rightarrow H \\
 u_1 &\mapsto u_4 u_3 u_1, & u_5 &\mapsto u_5 u_2 u_3^{-1}, \\
 u_2 &\mapsto u_3 u_2^{-1} u_4 u_3, & u_6 &\mapsto u_6 u_2 u_4 u_3 u_1, \\
 u_3 &\mapsto u_4 u_3 u_5 u_2 u_3^{-1}, & u_7 &\mapsto u_7 u_4^{-1} u_3 u_2^{-1}, \\
 u_4 &\mapsto u_4^{-1} u_5 u_2 u_3^{-1}, & u_8 &\mapsto u_8 u_2^3 u_3^{-1} u_4^{-1}.
 \end{aligned}$$

Now, Bob writes the ciphertext units

$$\begin{aligned}
 c_1 &= x y z x^3 y^2 z y^{-2} x^2, \\
 c_2 &= y z y^{-2} x^5 y z^{-1} y x^{-1} z^{-1} x y x, \\
 c_3 &= x^5 (z^2 y^{-3})^2, \\
 c_4 &= x^{-1} z x^{-1} y z^{-1} y^{-2} x z x^{-1}
 \end{aligned}$$

with the help of the algorithm given in Theorem 4.3.10 in letters of the Nielsen reduced set  $U$ . Step 1. and Step 2. are for all ciphertext units  $c_i$  equal.

Step 1: The Nielsen reduced set is  $U = \{u_1, u_2, \dots, u_8\}$  with

$$\begin{aligned}
 u_1 &:= x y z, & u_2 &:= y z y^{-1}, & u_3 &:= x^{-1} z x^{-1}, & u_4 &:= y^{-1} x^2, \\
 u_5 &:= z^{-1} x y x, & u_6 &:= z^{-1} y x^{-1}, & u_7 &:= x^3 y, & u_8 &:= y^3 z^{-2}.
 \end{aligned}$$

Step 2: Write each  $u \in U$  as  $u \equiv \ell(u)m(u)r(u)$  as in Corollary 4.2.10 with a stable part  $m(u)$ :

C.9. Example for decryption where Bob uses an algorithm to solve a constructive membership problem for a cryptosystem based on  $\text{Aut}(F_U)$

---

$$\begin{array}{lll}
 \ell(u_1) = x, & m(u_1) = y, & r(u_1) = z; \\
 \ell(u_2) = y, & m(u_2) = z, & r(u_2) = y^{-1}; \\
 \ell(u_3) = x^{-1}, & m(u_3) = z, & r(u_3) = x^{-1}; \\
 \ell(u_4) = y^{-1}, & m(u_4) = x, & r(u_4) = x; \\
 \ell(u_5) = z^{-1}, & m(u_5) = xy, & r(u_5) = x; \\
 \ell(u_6) = z^{-1}, & m(u_6) = y, & r(u_6) = x^{-1}; \\
 \ell(u_7) = x, & m(u_7) = x^2, & r(u_7) = y; \\
 \ell(u_8) = y, & m(u_8) = y^2z^{-2}, & r(u_8) = 1.
 \end{array}$$

- Bob uses the algorithm to write the ciphertext unit  $c_1 = xyzx^3y^2zy^{-2}x^2$  in letters of  $U^{\pm 1}$ . Table C.2 (page 326) shows the steps which were done in the algorithm without Step 1. and Step 2.

Table C.2.: Write the element  $c_1$  as a word in  $U$ 

	Action	Used element in $U$
Step 3 :	$c_1 = xyzx^3y^2zy^{-2}x^2 \neq 1$	
Step 4 :	$c_1 = \ell(u_1)m(u_1)zx^3y^2zy^{-2}x^2$	$u_1$
Step 5 :	$c'_1 := u_1^{-1}c_1 = x^3y^2zy^{-2}x^2$	
Step 3 :	$c'_1 = x^3y^2zy^{-2}x^2 \neq 1$	
Step 4 :	$c'_1 = \ell(u_7)m(u_7)y^2zy^{-2}x^2$	$u_7$
Step 5 :	$c''_1 := u_7^{-1}c'_1 = yzy^{-2}x^2$	
Step 3 :	$c''_1 = yzy^{-2}x^2 \neq 1$	
Step 4 :	$c''_1 = \ell(u_2)m(u_2)y^{-2}x^2$	$u_2$
Step 5 :	$c'''_1 := u_2^{-1}c''_1 = y^{-1}x^2$	
Step 3 :	$c'''_1 = y^{-1}x^2 \neq 1$	
Step 4 :	$c'''_1 = \ell(u_4)m(u_4)x$	$u_4$
Step 5 :	$c''''_1 := u_4^{-1}c'''_1 = 1$	
Step 3 :	$c''''_1 = 1$	

It is  $c''''_1 = 1$  and hence the algorithm stops. At the third column Bob gets the elements of  $U^{\pm 1}$  to write  $c_1$ . Hence, he knows  $c_1 = u_1u_7u_2u_4$ . Bob knows that this ciphertext unit was encrypted with the automorphism

$$\begin{aligned}
 f_{u_1} : H &\rightarrow H \\
 u_1 &\mapsto u_1u_7u_2u_4, & u_5 &\mapsto u_5^{-1}u_1u_7, \\
 u_2 &\mapsto u_2u_4u_3u_4^2, & u_6 &\mapsto u_6u_3u_4^2, \\
 u_3 &\mapsto u_3u_4^2, & u_7 &\mapsto u_6^{-1}u_4^{-1}u_7u_8, \\
 u_4 &\mapsto u_4u_6u_5^{-1}u_1u_7, & u_8 &\mapsto u_8u_1u_7.
 \end{aligned}$$

It is  $f_{u_1}(u_1) = u_1u_7u_2u_4 = c_1$ . Hence, he knows, that  $s_1 \hat{=} u_1$  was encrypted with  $c_1$  and thus the first letter is  $a_1 = L$ .

- Bob uses the algorithm to write the ciphertext unit  $c_2 = yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx$  in letters of  $U^{\pm 1}$ . Table C.3 (page 327) shows the steps which were done in the algorithm

C.9. Example for decryption where Bob uses an algorithm to solve a constructive membership problem for a cryptosystem based on  $\text{Aut}(F_U)$

without Step 1. and Step 2.

Table C.3.: Write the element  $c_2$  as a word in  $U$

	Action	Used element in $U$
Step 3 :	$c_2 = yzy^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx \neq 1$	
Step 4 :	$c_2 = \ell(u_2)m(u_2)y^{-2}x^5yz^{-1}yx^{-1}z^{-1}xyx$	$u_2$
Step 5 :	$c'_2 := u_2^{-1}c_2 = y^{-1}x^5yz^{-1}yx^{-1}z^{-1}xyx$	
Step 3 :	$c'_2 = y^{-1}x^5yz^{-1}yx^{-1}z^{-1}xyx \neq 1$	
Step 4 :	$c'_2 = \ell(u_4)m(u_4)x^4yz^{-1}yx^{-1}z^{-1}xyx$	$u_4$
Step 5 :	$c''_2 := u_4^{-1}c'_2 = x^3yz^{-1}yx^{-1}z^{-1}xyx$	
Step 3 :	$c''_2 = x^3yz^{-1}yx^{-1}z^{-1}xyx \neq 1$	
Step 4 :	$c''_2 = \ell(u_7)m(u_7)yz^{-1}yx^{-1}z^{-1}xyx$	$u_7$
Step 5 :	$c'''_2 := u_7^{-1}c''_2 = z^{-1}yx^{-1}z^{-1}xyx$	
Step 3 :	$c'''_2 = z^{-1}yx^{-1}z^{-1}xyx \neq 1$	
Step 4 :	$c'''_2 = \ell(u_6)m(u_6)x^{-1}z^{-1}xyx$	$u_6$
Step 5 :	$c''''_2 := u_6^{-1}c'''_2 = z^{-1}xyx$	
Step 3 :	$c''''_2 = z^{-1}xyx \neq 1$	
Step 4 :	$c''''_2 = \ell(u_5)m(u_5)x$	$u_5$
Step 5 :	$c_2'''' := u_5^{-1}c''''_2 = 1$	
Step 3 :	$c_2'''' = 1$	

It is  $c_2'''' = 1$  and hence the algorithm stops. At the third column Bob gets the elements of  $U^{\pm 1}$  to write  $c_2$ . Hence, he knows  $c_2 = u_2u_4u_7u_6u_5$ . Bob knows that this ciphertext

unit was encrypted with the automorphism

$$\begin{aligned}
 f_{u_2} : H &\rightarrow H \\
 u_1 &\mapsto u_3^{-1}u_1^{-1}u_2u_4, & u_5 &\mapsto u_5u_2^{-1}u_6u_5, \\
 u_2 &\mapsto u_2^{-1}u_8u_6u_5, & u_6 &\mapsto u_6u_5u_2u_4, \\
 u_3 &\mapsto u_3u_5u_4^{-2}, & u_7 &\mapsto u_7u_6u_5, \\
 u_4 &\mapsto u_2u_4u_7u_6u_5, & u_8 &\mapsto u_8u_4^{-1}u_2^{-1}.
 \end{aligned}$$

It is  $f_{u_2}(u_4) = u_2u_4u_7u_6u_5 = c_2$ . Hence, he knows, that  $s_2 \hat{=} u_4$  was encrypted with  $c_2$  and thus the second letter is  $a_4 = \mathbf{O}$ .

- Bob uses the algorithm to write the ciphertext unit  $c_3 = x^5(z^2y^{-3})^2$  in letters of  $U^{\pm 1}$ . Table C.4 (page 328) shows the steps which were done in the algorithm without Step 1. and Step 2.

Table C.4.: Write the element  $c_3$  as a word in  $U$

	Action	Used element in $U$
Step 3 :	$c_3 = x^5(z^2y^{-3})^2 \neq 1$	
Step 4 :	$c_3 = \ell(u_7)m(u_7)x^2(z^2y^{-3})^2$	$u_7$
Step 5 :	$c'_3 := u_7^{-1}c_3 = y^{-1}x^2(z^2y^{-3})^2$	
Step 3 :	$c'_3 = y^{-1}x^2(z^2y^{-3})^2 \neq 1$	
Step 4 :	$c'_3 = \ell(u_4)m(u_4)x(z^2y^{-3})^2$	$u_4$
Step 5 :	$c''_3 := u_4^{-1}c'_3 = (z^2y^{-3})^2$	
Step 3 :	$c''_3 = (z^2y^{-3})^2 \neq 1$	
Step 4 :	$c''_3 = r(u_8)^{-1}m(u_8)^{-1}y^{-1}z^2y^{-3}$	$u_8^{-1}$
Step 6 :	$c'''_3 := u_8c''_3 = z^2y^{-3}$	
Step 3 :	$c'''_3 = z^2y^{-3} \neq 1$	
Step 4 :	$c'''_3 = r(u_8)^{-1}m(u_8)^{-1}y^{-1}$	$u_8^{-1}$
Step 6 :	$c''''_3 := u_8c'''_3 = 1$	
Step 3 :	$c''''_3 = 1$	



C.9. Example for decryption where Bob uses an algorithm to solve a constructive membership problem for a cryptosystem based on  $\text{Aut}(F_U)$

---

It is  $c_3''' = 1$  and hence the algorithm stops. At the third column Bob gets the elements of  $U^{\pm 1}$  to write  $c_3$ . Hence, he knows  $c_3 = u_7 u_4 u_8^{-2}$ . Bob knows that this ciphertext unit was encrypted with the automorphism

$$\begin{aligned}
 f_{u_3} : H &\rightarrow H \\
 u_1 &\mapsto u_1 u_2^{-1} u_7^{-1} u_3^{-1} u_8, & u_5 &\mapsto u_5^{-1} u_6 u_3, \\
 u_2 &\mapsto u_2^{-1} u_3 u_7, & u_6 &\mapsto u_6 u_3^2 u_7, \\
 u_3 &\mapsto u_3 u_7 u_4 u_8^{-2}, & u_7 &\mapsto u_7 u_4 u_8^{-2}, \\
 u_4 &\mapsto u_4 u_8^{-2}, & u_8 &\mapsto u_7^{-1} u_3^{-1} u_8.
 \end{aligned}$$

It is  $f_{u_3}(u_7) = u_7 u_4 u_8^{-2} = c_3$ . Hence, he knows, that  $s_3 \hat{=} u_7$  was encrypted with  $c_3$  and thus the third letter is  $a_7 = V$ .

- Bob uses the algorithm to write the ciphertext unit  $c_4 = x^{-1} z x^{-1} y z^{-1} y^{-2} x z x^{-1}$  in letters of  $U^{\pm 1}$ . The Table C.5 (page 330) shows the steps which were done in the algorithm.

Table C.5.: Write the element  $c_4$  as a word in  $U$ 

	Action	Used element in $U$
Step 3 :	$c_4 = x^{-1}zx^{-1}yz^{-1}y^{-2}xzx^{-1} \neq 1$	
Step 4 :	$c_4 = \ell(u_3)m(u_3)x^{-1}yz^{-1}y^{-2}xzx^{-1}$	$u_3$
Step 5 :	$c'_4 := u_3^{-1}c_4 = yz^{-1}y^{-2}xzx^{-1}$	
Step 3 :	$c'_4 = yz^{-1}y^{-2}xzx^{-1} \neq 1$	
Step 4 :	$c'_4 = r(u_2)^{-1}m(u_2)^{-1}yz^{-1}y^{-2}xzx^{-1}$	$u_2^{-1}$
Step 6 :	$c''_4 := u_2c'_4 = y^{-1}xzx^{-1}$	
Step 3 :	$c''_4 = y^{-1}xzx^{-1} \neq 1$	
Step 4 :	$c''_4 = \ell(u_4)m(u_4)zx^{-1}$	$u_4$
Step 5 :	$c'''_4 := u_4^{-1}c''_4 = x^{-1}zx^{-1}$	
Step 3 :	$c'''_4 = x^{-1}zx^{-1} \neq 1$	
Step 4 :	$c'''_4 = \ell(u_3)m(u_3)x^{-1}$	$u_3$
Step 5 :	$c''''_4 := u_3^{-1}c'''_4 = 1$	
Step 3 :	$c''''_4 = 1$	

It is  $c''''_4 = 1$  and hence the algorithm stops. At the third column Bob gets the elements of  $U^{\pm 1}$  to write  $c_4$ . Hence, he knows  $c_4 = u_3u_2^{-1}u_4u_3$ . Bob knows that this ciphertext unit was encrypted with the automorphism

$$\begin{aligned}
 f_{u_4} : H &\rightarrow H \\
 u_1 &\mapsto u_4u_3u_1, & u_5 &\mapsto u_5u_2u_3^{-1}, \\
 u_2 &\mapsto u_3u_2^{-1}u_4u_3, & u_6 &\mapsto u_6u_2u_4u_3u_1, \\
 u_3 &\mapsto u_4u_3u_5u_2u_3^{-1}, & u_7 &\mapsto u_7u_4^{-1}u_3u_2^{-1}, \\
 u_4 &\mapsto u_4^{-1}u_5u_2u_3^{-1}, & u_8 &\mapsto u_8u_2^3u_3^{-1}u_4^{-1}.
 \end{aligned}$$

It is  $f_{u_4}(u_2) = u_3u_2^{-1}u_4u_3 = c_4$ . Hence, he knows, that  $s_4 \hat{=} u_2$  was encrypted with  $c_4$  and thus the fourth letter is  $a_2 = \mathbf{E}$ .

All together Bob reconstructs the correct message  $S = \mathbf{LOVE}$  from Alice.

## C.10. Example 9.0.7 calculated with GAP and Maple 16

For the private key cryptosystem in Example 9.0.7 we first present the calculations, which were done in GAP. After this we show the matrix multiplications, which were executed with the program Maple 16.

First, the free group  $F$  with generating set  $X = \{a, b, c, d\}$  is defined in GAP:

```
LoadPackage("FGA");;
F:=FreeGroup("a", "b", "c", "d");;
AssignGeneratorVariables(F);;
```

Alice and Bob also define the abstract groups  $G_1$  and  $G_2$ :

```
G1:=FreeGroup("x1", "x2", "x3", "x4", "x5");;
AssignGeneratorVariables(G1);;

G2:=FreeGroup("y1", "y2", "y3", "y4", "y5");;
AssignGeneratorVariables(G2);;
```

After this they choose the automorphisms  $g_{1_1}$  and  $g_{2_2}$ .

The automorphism  $g_{1_1}$  is describable with Nielsen transformations, as follows:

$$\begin{aligned}
 (x_1, x_2, x_3, x_4, x_5) &\xrightarrow{(N1)_1(N1)_4} (x_1^{-1}, x_2, x_3, x_4^{-1}, x_5) \\
 &\xrightarrow{(N2)_{2,3}} (x_1^{-1}, x_2x_3, x_3, x_4^{-1}, x_5) \\
 &\xrightarrow{(N2)_{5,3}} (x_1^{-1}, x_2x_3, x_3, x_4^{-1}, x_5x_3) \\
 &\xrightarrow{(N2)_{1,3}} (x_1^{-1}x_3, x_2x_3, x_3, x_4^{-1}, x_5x_3) \\
 &\xrightarrow{(N2)_{4,2}} (x_1^{-1}x_3, x_2x_3, x_3, x_4^{-1}x_2x_3, x_5x_3) \\
 &\xrightarrow{(N1)_5} (x_1^{-1}x_3, x_2x_3, x_3, x_4^{-1}x_2x_3, x_3^{-1}x_5^{-1}) \\
 &\xrightarrow{(N2)_{1,2}} (x_1^{-1}x_3x_2x_3, x_2x_3, x_3, x_4^{-1}x_2x_3, x_3^{-1}x_5^{-1}) \\
 &\xrightarrow{(N2)_{2,4}} (x_1^{-1}x_3x_2x_3, x_2x_3x_4^{-1}x_2x_3, x_3, x_4^{-1}x_2x_3, x_3^{-1}x_5^{-1}) \\
 &\xrightarrow{(N2)_{3,1}} (x_1^{-1}x_3x_2x_3, x_2x_3x_4^{-1}x_2x_3, x_3x_1^{-1}x_3x_2x_3, x_4^{-1}x_2x_3, x_3^{-1}x_5^{-1})
 \end{aligned}$$

Hence, the automorphism is

$$\begin{aligned}
 g_{1_1} : G_1 &\rightarrow G_1 \\
 x_1 &\mapsto x_1^{-1}x_3x_2x_3, \\
 x_2 &\mapsto x_2x_3x_4^{-1}x_2x_3, \\
 x_3 &\mapsto x_3x_1^{-1}x_3x_2x_3, \\
 x_4 &\mapsto x_4^{-1}x_2x_3, \\
 x_5 &\mapsto x_3^{-1}x_5^{-1}x_3.
 \end{aligned}$$

For decryption Bob needs the inverse automorphism  $g_{1_1}^{-1}$ , which is described with Nielsen transformations in the following way:

$$\begin{aligned}
 (x_{11}, x_{21}, x_{31}, x_{41}, x_{51}) &\xrightarrow{(N1)_1} (x_{11}^{-1}, x_{21}, x_{31}, x_{41}, x_{51}) \\
 &\xrightarrow{(N2)_{3,1}} (x_{11}^{-1}, x_{21}, x_{31}x_{11}^{-1}, x_{41}, x_{51}) \\
 &\xrightarrow{(N1)_1(N1)_4} (x_{11}, x_{21}, x_{31}x_{11}^{-1}, x_{41}^{-1}, x_{51}) \\
 &\xrightarrow{(N2)_{2,4}} (x_{11}, x_{21}x_{41}^{-1}, x_{31}x_{11}^{-1}, x_{41}^{-1}, x_{51}) \\
 &\xrightarrow{(N1)_4(N1)_2} (x_{11}, x_{41}x_{21}^{-1}, x_{31}x_{11}^{-1}, x_{41}, x_{51}) \\
 &\xrightarrow{(N2)_{1,2}} (x_{11}x_{41}x_{21}^{-1}, x_{41}x_{21}^{-1}, x_{31}x_{11}^{-1}, x_{41}, x_{51}) \\
 &\xrightarrow{(N2)_{4,2}} (x_{11}x_{41}x_{21}^{-1}, x_{41}x_{21}^{-1}, x_{31}x_{11}^{-1}, x_{41}^2x_{21}^{-1}, x_{51}) \\
 &\xrightarrow{(N1)_5(N1)_3} (x_{11}x_{41}x_{21}^{-1}, x_{41}x_{21}^{-1}, x_{11}x_{31}^{-1}, x_{41}^2x_{21}^{-1}, x_{51}^{-1}) \\
 &\xrightarrow{(N2)_{5,3}} (x_{11}x_{41}x_{21}^{-1}, x_{41}x_{21}^{-1}, x_{11}x_{31}^{-1}, x_{41}^2x_{21}^{-1}, x_{51}^{-1}x_{11}x_{31}^{-1}) \\
 &\xrightarrow{(N1)_5} (x_{11}x_{41}x_{21}^{-1}, x_{41}x_{21}^{-1}, x_{11}x_{31}^{-1}, x_{41}^2x_{21}^{-1}, x_{31}x_{11}^{-1}x_{51}) \\
 &\xrightarrow{(N2)_{5,3}} (x_{11}x_{41}x_{21}^{-1}, x_{41}x_{21}^{-1}, x_{11}x_{31}^{-1}, x_{41}^2x_{21}^{-1}, x_{31}x_{11}^{-1}x_{51}x_{11}x_{31}^{-1}) \\
 &\xrightarrow{(N2)_{1,3}} (x_{11}x_{41}x_{21}^{-1}x_{11}x_{31}^{-1}, x_{41}x_{21}^{-1}, x_{11}x_{31}^{-1}, x_{41}^2x_{21}^{-1}, x_{31}x_{11}^{-1}x_{51}x_{11}x_{31}^{-1}) \\
 &\xrightarrow{(N1)_2} (x_{11}x_{41}x_{21}^{-1}x_{11}x_{31}^{-1}, x_{21}x_{41}^{-1}, x_{11}x_{31}^{-1}, x_{41}^2x_{21}^{-1}, x_{31}x_{11}^{-1}x_{51}x_{11}x_{31}^{-1}) \\
 &\xrightarrow{(N2)_{2,3}} (x_{11}x_{41}x_{21}^{-1}x_{11}x_{31}^{-1}, x_{21}x_{41}^{-1}x_{11}x_{31}^{-1}, x_{11}x_{31}^{-1}, x_{41}^2x_{21}^{-1}, x_{31}x_{11}^{-1}x_{51}x_{11}x_{31}^{-1}) \\
 &\xrightarrow{(N1)_1(N1)_3} ((x_{11}x_{41}x_{21}^{-1}x_{11}x_{31}^{-1})^{-1}, x_{21}x_{41}^{-1}x_{11}x_{31}^{-1}, x_{31}x_{11}^{-1}, x_{41}^2x_{21}^{-1}, x_{31}x_{11}^{-1}x_{51}x_{11}x_{31}^{-1}) \\
 &\xrightarrow{(N1)_4(N1)_5} ((x_{11}x_{41}x_{21}^{-1}x_{11}x_{31}^{-1})^{-1}, x_{21}x_{41}^{-1}x_{11}x_{31}^{-1}, x_{31}x_{11}^{-1}, x_{21}x_{41}^{-2}, (x_{31}x_{11}^{-1}x_{51}x_{11}x_{31}^{-1})^{-1})
 \end{aligned}$$

Hence, the inverse automorphism of  $g_{1_1}$  is

$$\begin{aligned}
 g_{1_1}^{-1} : G_1 &\rightarrow G_1 \\
 x_1 &\mapsto (x_1x_4x_2^{-1}x_1x_3^{-1})^{-1}, \\
 x_2 &\mapsto x_2x_4^{-1}x_1x_3^{-1}, \\
 x_3 &\mapsto x_3x_1^{-1}, \\
 x_4 &\mapsto x_2x_4^{-2}, \\
 x_5 &\mapsto (x_3x_1^{-1}x_5x_1x_3^{-1})^{-1}.
 \end{aligned}$$

In GAP they define

$$\begin{aligned}
 x11 &:= x1^{-1} * x3 * x2 * x3; \\
 x21 &:= x2 * x3 * x4^{-1} * x2 * x3; \\
 x31 &:= x3 * x1^{-1} * x3 * x2 * x3; \\
 x41 &:= x4^{-1} * x2 * x3; \\
 x51 &:= x3^{-1} * x5^{-1} * x3;
 \end{aligned}$$

to apply the automorphism  $g_{1_1}$  on  $x_1, x_2, x_3, x_4$  and  $x_5$  and they define

$$\begin{aligned}
 x11i &:= (x11 * x41 * x21^{-1} * x11 * x31^{-1})^{-1}; \\
 x21i &:= x21 * x41^{-1} * x11 * x31^{-1};
 \end{aligned}$$

```
x31i:=x31*x11^-1;
x41i:=x21*x41^-2;
x51i:=(x31*x11^-1*x51*x11*x31^-1)^-1;
```

to proof if  $g_{1_1}^{-1}$  is the correct inverse automorphism for  $g_{1_1}$ :

```
gap> x11i;
      x1
gap> x21i;
      x2
gap> x31i;
      x3
gap> x41i;
      x4
gap> x51i;
      x5
```

The automorphism  $g_{2_2}$  is describable with Nielsen transformations, as follows:

$$\begin{aligned}
 (y_1, y_2, y_3, y_4) &\xrightarrow{[(N2)_{3,1}]^2} (y_1, y_2, y_3y_1^2, y_4) \\
 &\xrightarrow{(N1)_2} (y_1, y_2^{-1}, y_3y_1^2, y_4) \\
 &\xrightarrow{[(N2)_{2,1}]^3} (y_1, y_2^{-1}y_1^3, y_3y_1^2, y_4) \\
 &\xrightarrow{(N2)_{2,4}} (y_1, y_2^{-1}y_1^3y_4, y_3y_1^2, y_4) \\
 &\xrightarrow{(N2)_{4,2}} (y_1, y_2^{-1}y_1^3y_4, y_3y_1^2, y_4y_2^{-1}y_1^3y_4) \\
 &\xrightarrow{(N2)_{1,3}} (y_1y_3y_1^2, y_2^{-1}y_1^3y_4, y_3y_1^2, y_4y_2^{-1}y_1^3y_4)
 \end{aligned}$$

Therefore, the automorphism is

$$\begin{aligned}
 g_{2_2} : G_2 &\rightarrow G_2 \\
 y_1 &\mapsto y_1y_3y_1^2, \\
 y_2 &\mapsto y_2^{-1}y_1^3y_4, \\
 y_3 &\mapsto y_3y_1^2, \\
 y_4 &\mapsto y_4y_2^{-1}y_1^3y_4.
 \end{aligned}$$

For decryption Bob needs the inverse automorphism  $g_{2_2}^{-1}$ , which is describable with Nielsen transformations in the following way

$$\begin{aligned}
 (y_{11}, y_{21}, y_{31}, y_{41}) &\xrightarrow{(N1)_2} (y_{11}, y_{21}^{-1}, y_{31}, y_{41}) \\
 &\xrightarrow{(N2)_{4,2}} (y_{11}, y_{21}^{-1}, y_{31}, y_{41}y_{21}^{-1}) \\
 &\xrightarrow{(N1)_3} (y_{11}, y_{21}^{-1}, y_{31}^{-1}, y_{41}y_{21}^{-1}) \\
 &\xrightarrow{(N2)_{1,3}} (y_{11}y_{31}^{-1}, y_{21}^{-1}, y_{31}^{-1}, y_{41}y_{21}^{-1}) \\
 &\xrightarrow{(N1)_3(N1)_1} (y_{31}y_{11}^{-1}, y_{21}^{-1}, y_{31}, y_{41}y_{21}^{-1}) \\
 &\xrightarrow{[(N2)_{3,1}]^2} (y_{31}y_{11}^{-1}, y_{21}^{-1}, y_{31}^2y_{11}^{-1}y_{31}y_{11}^{-1}, y_{41}y_{21}^{-1}) \\
 &\xrightarrow{(N1)_2(N1)_4} (y_{31}y_{11}^{-1}, y_{21}, y_{31}^2y_{11}^{-1}y_{31}y_{11}^{-1}, y_{21}y_{41}^{-1}) \\
 &\xrightarrow{(N2)_{2,4}} (y_{31}y_{11}^{-1}, y_{21}y_{41}^{-1}, y_{31}^2y_{11}^{-1}y_{31}y_{11}^{-1}, y_{21}y_{41}^{-1}) \\
 &\xrightarrow{[(N2)_{2,1}]^3} (y_{31}y_{11}^{-1}, y_{21}y_{41}^{-1}(y_{31}y_{11}^{-1})^3, y_{31}^2y_{11}^{-1}y_{31}y_{11}^{-1}, y_{21}y_{41}^{-1}) \\
 &\xrightarrow{(N1)_1(N1)_2(N1)_4} (y_{11}y_{31}^{-1}, (y_{21}y_{41}^{-1}(y_{31}y_{11}^{-1})^3)^{-1}, y_{31}^2y_{11}^{-1}y_{31}y_{11}^{-1}, y_{41}y_{21}^{-1});
 \end{aligned}$$

hence it is

$$\begin{aligned}
 g_{2_2}^{-1} : G_2 &\rightarrow G_2 \\
 y_1 &\mapsto y_1y_3^{-1}, \\
 y_2 &\mapsto (y_2^2y_4^{-1}(y_3y_1^{-1})^3)^{-1}, \\
 y_3 &\mapsto y_3^2y_1^{-1}y_3y_1^{-1}, \\
 y_4 &\mapsto y_4y_2^{-1}.
 \end{aligned}$$

In GAP they define

```

y11:=y1*y3*(y1^2);
y21:=(y2^(-1))*(y1^3)*y4;
y31:=y3*(y1^2);
y41:=y4*(y2^(-1))*(y1^3)*y4;

```

to apply the automorphism  $g_{2_2}$  on  $y_1, y_2, y_3$  and  $y_4$  and they define

```

y11i:=y11*y31^-1;
y21i:=(y21^2*y41^-1*(y31*y11^-1)^3)^-1;
y31i:=y31^2*y11^-1*y31*y11^-1;
y41i:=y41*y21^-1;

```

to proof if  $g_{2_2}^{-1}$  is the correct inverse automorphism for  $g_{2_2}$ :

```

gap> y11i;
y1
gap> y21i;
y2
gap> y31i;
y3
gap> y41i;
y4

```

For their alphabet  $A$  they need a Nielsen reduced set  $U \subset F$  with  $2|A|$  elements. They choose

$$U = \{ba^2, cd, d^2c^{-2}, a^{-1}b, a^4b^{-1}, b^3a^{-2}, bc^3, bc^{-1}bab^{-1}, c^2ba, c^2dab^{-1}, dabd^{-1}a, a^{-1}d^3c^{-1}, a^{-1}c^{-1}bac^{-2}, a^2db^2d^{-1}\}.$$

They prove in GAP, if this is a Nielsen reduced set. If the set  $U$  is used as free generating set for a subgroup  $FU$  of  $F$ , then the operation

▷ `FreeGeneratorsOfGroup(FU)`

gives a Nielsen reduced generating set for  $FU$ :

```
FU:=Group( b*a^2, c*d, d^2*c^-2, a^-1*b, a^4*b^-1, b^3*a^-2, b*c^3,\
           b*c^-1*b*a*b^-1, c^2*b*a, c^2*d*a*b^-1, d*a*b*d^-1*a,\
           a^-1*d^3*c^-1, a^-1*c^-1*b*a*c^-2, a^2*d*b^2*d^-1 );;

gap> FU;
Group([ b*a^2, c*d, d^2*c^-2, a^-1*b, a^4*b^-1, b^3*a^-2, b*c^3,\
        b*c^-1*b*a*b^-1, c^2*b*a, c^2*d*a*b^-1, d*a*b*d^-1*a,\
        a^-1*d^3*c^-1, a^-1*c^-1*b*a*c^-2, a^2*d*b^2*d^-1 ])
gap> FreeGeneratorsOfGroup(FU);
[ b*a^2, c*d, d^2*c^-2, a^-1*b, a^4*b^-1, b^3*a^-2, b*c^3,\
  b*c^-1*b*a*b^-1, c^2*b*a, c^2*d*a*b^-1, d*a*b*d^-1*a,\
  a^-1*d^3*c^-1, a^-1*c^-1*b*a*c^-2, a^2*d*b^2*d^-1 ]
```

For the ephemeral matrices from Alice she uses an abstract group  $F_N$  of rank 3 with free generating set  $N = \{N_1, N_2, N_3\}$  and the Nielsen reduced set is  $\{N_1N_2^2, N_2N_3, N_3N_1^2, N_1^{-1}N_2N_1N_2\}$ . If this set is used as free generating set for a subgroup  $F_{N'} =: H$  of  $F$ , then the operation

▷ `FreeGeneratorsOfGroup(H)`

gives a Nielsen reduced generating set for  $H$ :

```
N:=FreeGroup("N1", "N2", "N3");;
AssignGeneratorVariables(N);;
H:=Group(N1*N2^2, N2*N3, N3*N1^2, N1^-1*N2*N1*N2 );;

gap> H;
Group([ N1*N2^2, N2*N3, N3*N1^2, N1^-1*N2*N1*N2 ])
gap> FreeGeneratorsOfGroup(H);
[ N1*N2^2, N2*N3, N3*N1^2, N1^-1*N2*N1*N2 ]
```

We take a closer look at the calculations which were executed with Maple 16.

```
> restart; with(LinearAlgebra):
```

For the faithful representation  $\varphi$  they need matrices in  $SL(2, \mathbb{Q})$  and generate them with Theorem 4.2.18. For this they choose rational numbers  $r_i =: r[i]$ ,  $1 \leq i \leq 4$ , with the properties (4.1). Hence, these rational numbers were chosen as follows and the inequalities (4.1) were proved:

```
> r[1] := 7/2;
> r[2] := 15/2;
> r[3] := 23/2;
> r[4] := 35/2;
```

$$r_1 := \frac{7}{2}$$

$$r_2 := \frac{15}{2}$$

$$r_3 := \frac{23}{2}$$

$$r_4 := \frac{35}{2}$$

```
> r[1]-2;
> r[2]-r[1]-3;
> r[3]-r[2]-3;
> r[4]-r[3]-3;
```

$$\frac{3}{2}$$

$$1$$

$$1$$

$$3$$

All results are greater than 0, hence they can generate with the numbers  $r_1, r_2, r_3$  and  $r_4$  matrices which generate a free subgroup of  $SL(2, \mathbb{Q})$  of rank 4. The matrices for Alice and Bob are the following:

```
> M[1] := Matrix([[ -r[1], r[1]^2-1], [1, -r[1]]]);
> M[2] := Matrix([[ -r[2], r[2]^2-1], [1, -r[2]]]);
> M[3] := Matrix([[ -r[3], r[3]^2-1], [1, -r[3]]]);
> M[4] := Matrix([[ -r[4], r[4]^2-1], [1, -r[4]]]);
```

$$M_1 := \begin{bmatrix} -\frac{7}{2} & \frac{45}{4} \\ 1 & -\frac{7}{2} \end{bmatrix}$$

$$M_2 := \begin{bmatrix} -\frac{15}{2} & \frac{221}{4} \\ 1 & -\frac{15}{2} \end{bmatrix}$$

$$M_3 := \begin{bmatrix} -\frac{23}{2} & \frac{525}{4} \\ 1 & -\frac{23}{2} \end{bmatrix}$$

$$M_4 := \begin{bmatrix} -\frac{35}{2} & \frac{1221}{4} \\ 1 & -\frac{35}{2} \end{bmatrix}$$

Because of the faithful representation  $\varphi$  they define:

```
> a:=M[1];
> b:=M[2];
> c:=M[3];
> d:=M[4];
```



$$a := \begin{bmatrix} \frac{-7}{2} & \frac{45}{4} \\ 1 & \frac{-7}{2} \end{bmatrix}$$

$$b := \begin{bmatrix} \frac{-15}{2} & \frac{221}{4} \\ 1 & \frac{-15}{2} \end{bmatrix}$$

$$c := \begin{bmatrix} \frac{-23}{2} & \frac{525}{4} \\ 1 & \frac{-23}{2} \end{bmatrix}$$

$$d := \begin{bmatrix} \frac{-35}{2} & \frac{1221}{4} \\ 1 & \frac{-35}{2} \end{bmatrix}$$

Now, they are able to calculate the elements of the Nielsen reduced set  $U'$ :

```
> V1:=b.a.a;
> V2:=c.d;
> V3:=d.d.MatrixInverse(c).MatrixInverse(c);
> V4:=MatrixInverse(a).b;
> V5:=a.a.a.a.MatrixInverse(b);
> V6:=b.b.b.MatrixInverse(a).MatrixInverse(a);
> V7:=b.c.c.c;
> V8:=b.MatrixInverse(c).b.a.MatrixInverse(b);
> V9:=c.c.b.a;
> V10:=c.c.d.a.MatrixInverse(b);
> V11:=d.a.b.MatrixInverse(d).a;
> V12:=MatrixInverse(a).d.d.d.MatrixInverse(c);
> V13:=MatrixInverse(a).MatrixInverse(c).b.a.MatrixInverse(c).MatrixInve
> rse(c);
> V14:=a.a.d.b.b.MatrixInverse(d);
```

$$V1 := \begin{bmatrix} -563 & 1889 \\ 76 & -255 \end{bmatrix}$$

$$V2 := \begin{bmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{bmatrix}$$

$$V3 := \begin{bmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{bmatrix}$$

$$V4 := \begin{bmatrix} 15 & -109 \\ 4 & -29 \end{bmatrix}$$

$$V5 := \begin{bmatrix} -4575 & -33209 \\ 1364 & 9901 \end{bmatrix}$$

$$\begin{aligned}
 V6 &:= \begin{bmatrix} \frac{95009}{2} & \frac{638869}{4} \\ -6391 & \frac{-42975}{2} \end{bmatrix} \\
 V7 &:= \begin{bmatrix} \frac{149079}{2} & \frac{-3415829}{4} \\ -10009 & \frac{229335}{2} \end{bmatrix} \\
 V8 &:= \begin{bmatrix} \frac{10733}{2} & \frac{155745}{4} \\ -691 & \frac{-10027}{2} \end{bmatrix} \\
 V9 &:= \begin{bmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{bmatrix} \\
 V10 &:= \begin{bmatrix} -647496 & \frac{-9392507}{2} \\ 56518 & 409922 \end{bmatrix} \\
 V11 &:= \begin{bmatrix} 563077 & -2011276 \\ -32264 & 115245 \end{bmatrix} \\
 V12 &:= \begin{bmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{bmatrix} \\
 V13 &:= \begin{bmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{bmatrix} \\
 V14 &:= \begin{bmatrix} \frac{3682603}{2} & \frac{128159475}{4} \\ -548633 & \frac{-19093157}{2} \end{bmatrix}
 \end{aligned}$$

Encryption:

To write the message in sequences of matrices Alice uses the assignment

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N}$$

between the alphabet elements  $a_i \in A$  and the matrices  $V_j \in U'$ .

For her first message sequence  $S'_1$  it is:

```

ILIK ≐ V2 V4 V9 V12
> s11:=V2;
> s12:=V4;
> s13:=V9;
> s14:=V12;

```

$$s11 := \begin{bmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{bmatrix}$$

$$s12 := \begin{bmatrix} 15 & -109 \\ 4 & -29 \end{bmatrix}$$

$$s13 := \begin{bmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{bmatrix}$$

$$s14 := \begin{bmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{bmatrix}$$

For the second message sequence  $S'_2$  it is:

```
EB ≐ V3 V1
> s21:=V3;
> s22:=V1;
```

$$s21 := \begin{bmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{bmatrix}$$

$$s22 := \begin{bmatrix} -563 & 1889 \\ 76 & -255 \end{bmatrix}$$

and for the third message sequence  $S'_3$  it is:

```
OB ≐ V13 V1
> s31:=V13;
> s32:=V1;
```

$$s31 := \begin{bmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{bmatrix}$$

$$s32 := \begin{bmatrix} -563 & 1889 \\ 76 & -255 \end{bmatrix}$$

To generate ephemeral keys  $P_1, P_2, P_3$  and  $P_4$ , Alice generates first matrices  $N_1, N_2, N_3$  also with the Theorem 4.2.18 under considerations of the matrices  $M_1, M_2, M_3$  and  $M_4$ . She proves the inequalities (4.1):

```
> r[1] := 7/2;
> r[2] := 15/2;
> r[3] := 23/2;
> r[4] := 35/2;
> r[5] := 43/2;
> r[6] := 55/2;
> r[7] := 63/2;
```

$$r_1 := \frac{7}{2}$$

$$r_2 := \frac{15}{2}$$

$$r_3 := \frac{23}{2}$$

```

r4 := 35/2
r5 := 43/2
r6 := 55/2
r7 := 63/2

> r[1]-2;
> r[2]-r[1]-3;
> r[3]-r[2]-3;
> r[4]-r[3]-3;
> r[5]-r[4]-3;
> r[6]-r[5]-3;
> r[7]-r[6]-3;

3
2
1
1
3
1
3
1

```

All results are greater than 0, hence she can generate with the numbers  $r_1, r_2, \dots, r_7$  matrices which generate a free subgroup of  $SL(2, \mathbb{Q})$  of rank 7. Now, her matrices for the generating set  $N$  are:

```

> N[1] := Matrix([[ -r[5], r[5]^2-1], [1, -r[5]]]);
> N[2] := Matrix([[ -r[6], r[6]^2-1], [1, -r[6]]]);
> N[3] := Matrix([[ -r[7], r[7]^2-1], [1, -r[7]]]);

```

$$N_1 := \begin{bmatrix} \frac{-43}{2} & \frac{1845}{4} \\ 1 & \frac{-43}{2} \end{bmatrix}$$

$$N_2 := \begin{bmatrix} \frac{-55}{2} & \frac{3021}{4} \\ 1 & \frac{-55}{2} \end{bmatrix}$$

$$N_3 := \begin{bmatrix} \frac{-63}{2} & \frac{3965}{4} \\ 1 & \frac{-63}{2} \end{bmatrix}$$

With the matrices in the generating set  $N$  she chooses the Nielsen reduced set  $\{P1, P2, P3, P4\}$  as follows:

```

> P1:=N[1].N[2].N[2];
> P2:=N[2].N[3];
> P3:=N[3].N[1].N[1];
> P4:=MatrixInverse(N[1]).N[2].N[1].N[2];

```

$$P1 := \begin{bmatrix} -57866 & \frac{3180525}{2} \\ 2694 & -74036 \end{bmatrix}$$

$$P2 := \begin{bmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{bmatrix}$$

$$P3 := \begin{bmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{bmatrix}$$

$$P4 := \begin{bmatrix} \frac{621893}{2} & \frac{-34178721}{4} \\ 14351 & \frac{-788719}{2} \end{bmatrix}$$

To encrypt her first message sequence  $S'_1$  she added the ephemeral matrix  $P_1$  to the sequence:

```
> y1:=s11;
> y2:=P1;
> y3:=s12;
> y4:=s13;
> y5:=s14;
```

$$y1 := \begin{bmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{bmatrix}$$

$$y2 := \begin{bmatrix} -57866 & \frac{3180525}{2} \\ 2694 & -74036 \end{bmatrix}$$

$$y3 := \begin{bmatrix} 15 & -109 \\ 4 & -29 \end{bmatrix}$$

$$y4 := \begin{bmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{bmatrix}$$

$$y5 := \begin{bmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{bmatrix}$$

The sequence  $S'_1$  is encrypted with the automorphism  $g_{1_1}$  as:

```
> q11:=MatrixInverse(y1).y3.y2.y3;
> q12:=y2.y3.MatrixInverse(y4).y2.y3;
> q13:=y3.MatrixInverse(y1).y3.y2.y3;
> q14:=MatrixInverse(y4).y2.y3;
> q15:=MatrixInverse(y3).MatrixInverse(y5).y3;
```

$$q11 := \begin{bmatrix} \frac{453037463005}{2} & \frac{-6566656978411}{4} \\ 12969541169 & \frac{-187990033891}{2} \end{bmatrix}$$

$$q12 := \begin{bmatrix} \frac{-515958453260453803}{2} & \frac{7478679920196901999}{4} \\ 12010438543010031 & \frac{-174088097591505391}{2} \end{bmatrix}$$

$$q13 := \begin{bmatrix} \frac{3968201970233}{2} & \frac{-57518027287927}{4} \\ 529958232109 & \frac{-7681602973983}{2} \end{bmatrix}$$

$$q14 := \begin{bmatrix} \frac{83406030953}{2} & \frac{-1208948133265}{4} \\ 12234456659 & \frac{-177335180327}{2} \end{bmatrix}$$

$$q15 := \begin{bmatrix} \frac{-65207575}{2} & \frac{991192539}{4} \\ -4494561 & \frac{68319905}{2} \end{bmatrix}$$

For her second message sequence  $S'_2$  she added the ephemeral keys  $P_2$  and  $P_3$  to the sequence:

```
> y1:=s21;
> y2:=s22;
> y3:=P2;
> y4:=P3;
```

$$y1 := \begin{bmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{bmatrix}$$

$$y2 := \begin{bmatrix} -563 & 1889 \\ 76 & -255 \end{bmatrix}$$

$$y3 := \begin{bmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{bmatrix}$$

$$y4 := \begin{bmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{bmatrix}$$

Thus, the sequence  $S'_2$  is encrypted with the automorphism  $g_{2_2}$  as:

```
> q21:=y1.y3.y1.y1;
> q22:=MatrixInverse(y2).y1.y1.y1.y4;
> q23:=y3.y1.y1;
> q24:=y4.MatrixInverse(y2).y1.y1.y1.y4;
```

$$q21 := \begin{bmatrix} \frac{-1104332496534507861}{2} & \frac{-25304312660337129571}{4} \\ 31604200843034185 & \frac{724168293536436571}{2} \end{bmatrix}$$

$$q22 := \begin{bmatrix} -480689945680474129277 & 10323650084255317045974 \\ -143263719821090419728 & 3076836797799093562123 \end{bmatrix}$$

$$q23 := \begin{bmatrix} \frac{22386390293811}{2} & \frac{512954405587601}{4} \\ -407276382779 & \frac{-9332197468925}{2} \end{bmatrix}$$

$$q24 := \begin{bmatrix} -186180075388817073675749582 & \frac{7997079898367833227219056023}{2} \\ 5914022532266907628279666 & -127013888603589241202576880 \end{bmatrix}$$

The ephemeral keys  $P_4$ ,  $P_2$  and  $P_3$  are added to the sequence  $S'_3$ :

```
> y1:=s31;
> y2:=P4;
> y3:=s32;
> y4:=P2;
> y5:=P3;
```

$$y1 := \begin{bmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{bmatrix}$$

$$y2 := \begin{bmatrix} \frac{621893}{2} & \frac{-34178721}{4} \\ 14351 & \frac{-788719}{2} \end{bmatrix}$$

$$y3 := \begin{bmatrix} -563 & 1889 \\ 76 & -255 \end{bmatrix}$$

$$y4 := \begin{bmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{bmatrix}$$

$$y5 := \begin{bmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{bmatrix}$$

Hence, the sequence  $S'_3$  is encrypted with the automorphism  $g_{3_1}$  as:

```
> q31:=MatrixInverse(y1).y3.y2.y3;
> q32:=y2.y3.MatrixInverse(y4).y2.y3;
> q33:=y3.MatrixInverse(y1).y3.y2.y3;
> q34:=MatrixInverse(y4).y2.y3;
> q35:=MatrixInverse(y3).MatrixInverse(y5).y3;
```

$$q31 := \begin{bmatrix} \frac{-1616087435846771117}{2} & \frac{10844781227098250059}{4} \\ 70532776776146599 & \frac{-473311354639843285}{2} \end{bmatrix}$$

$$q32 := \begin{bmatrix} \frac{5117735040480436319307}{2} & \frac{-34342644872543531950151}{4} \\ 118098476048874935309 & \frac{-792501759245893528165}{2} \end{bmatrix}$$

$$q33 := \begin{bmatrix} \frac{1176330057042013989893}{2} & \frac{-7893782128685642713947}{4} \\ -79397180640094685191 & \frac{532796082062893539917}{2} \end{bmatrix}$$

$$q34 := \begin{bmatrix} -3473922528580 & \frac{23311814068153}{2} \\ -110342647234 & 370228071430 \end{bmatrix}$$

$$q35 := \begin{bmatrix} 30697842540 & \frac{-205999121749}{2} \\ 9149177258 & -30697963178 \end{bmatrix}$$

Decryption:

For the decryption Bob applies the inverse automorphism  $g_{i_j}^{-1}$  on the ciphertext sequences  $C'_i$ .

He starts with  $g_{1_1}^{-1}$  and the ciphertext sequence  $C'_1$ :

```
> a11:=q11.q14.MatrixInverse(q12).q11.MatrixInverse(q13):
> a111:=MatrixInverse(a11);
> b11:=q12.MatrixInverse(q14).q11.MatrixInverse(q13);
> c11:=q13.MatrixInverse(q11);
> d11:=q12.MatrixInverse(q14).MatrixInverse(q14);
> e11:=q13.MatrixInverse(q11).q15.q11.MatrixInverse(q13):
> e111:=MatrixInverse(e11);
```

$$a111 := \begin{bmatrix} \frac{665}{2} & \frac{-23229}{4} \\ -29 & \frac{1013}{2} \end{bmatrix}$$

$$b11 := \begin{bmatrix} -57866 & \frac{3180525}{2} \\ 2694 & -74036 \end{bmatrix}$$

$$c11 := \begin{bmatrix} 15 & -109 \\ 4 & -29 \end{bmatrix}$$

$$d11 := \begin{bmatrix} \frac{109363}{2} & \frac{-745561}{4} \\ -4773 & \frac{32539}{2} \end{bmatrix}$$

$$e111 := \begin{bmatrix} \frac{729437}{2} & \frac{17021361}{4} \\ 102117 & \frac{2382893}{2} \end{bmatrix}$$

Next, he applies  $g_{2_2}^{-1}$  on the ciphertext sequence  $C'_2$ :

```
> a22:=q21.MatrixInverse(q23);
> b22:=q22.q22.MatrixInverse(q24).q23.MatrixInverse(q21).q23.MatrixInver
> se(q21).q23.MatrixInverse(q21):
> b222:=MatrixInverse(b22);
> c22:=q23.q23.MatrixInverse(q21).q23.MatrixInverse(q21);
> d22:=q24.MatrixInverse(q22);
```

$$a22 := \begin{bmatrix} -84596 & \frac{-1938405}{2} \\ 4842 & 55474 \end{bmatrix}$$

$$b222 := \begin{bmatrix} -563 & 1889 \\ 76 & -255 \end{bmatrix}$$

$$c22 := \begin{bmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{bmatrix}$$

$$d22 := \begin{bmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{bmatrix}$$



Finally, he applies  $g_{3_1}^{-1} = g_{1_1}^{-1}$  on the ciphertext sequence  $C'_3$  and gets the following:

```
> a33:=q31.q34.MatrixInverse(q32).q31.MatrixInverse(q33):
> a333:=MatrixInverse(a33);
> b33:=q32.MatrixInverse(q34).q31.MatrixInverse(q33);
> c33:=q33.MatrixInverse(q31);
> d33:=q32.MatrixInverse(q34).MatrixInverse(q34);
> e33:=q33.MatrixInverse(q31).q35.q31.MatrixInverse(q33):
> e333:=MatrixInverse(e33);
```

$$a333 := \begin{bmatrix} \frac{-843429}{2} & \frac{-19325129}{4} \\ -122869 & \frac{-2815245}{2} \end{bmatrix}$$

$$b33 := \begin{bmatrix} \frac{621893}{2} & \frac{-34178721}{4} \\ 14351 & \frac{-788719}{2} \end{bmatrix}$$

$$c33 := \begin{bmatrix} -563 & 1889 \\ 76 & -255 \end{bmatrix}$$

$$d33 := \begin{bmatrix} \frac{3243}{2} & \frac{-204199}{4} \\ -59 & \frac{3715}{2} \end{bmatrix}$$

$$e333 := \begin{bmatrix} -71714 & \frac{3080365}{2} \\ 2278 & -48924 \end{bmatrix}$$

With the plaintext alphabet  $A$  and the assignment

$$a_i \hat{=} V_j \iff j \equiv i \pmod{N},$$

for  $a_i \in A$  and  $V_j \in U'$ , he is able to reconstruct the message ILIKEBOB from Alice.

## C.11. Example 10.1.4 executed with GAP

For Example 10.1.4 we used the program GAP.

Alice defines the public parameters.

Let  $X = \{x, y, z\}$  be the free generating set for a free subgroup of rank 3:

```
LoadPackage("FGA");;
F:=FreeGroup("x", "y", "z");;
AssignGeneratorVariables(F);;
```

Additionally she defines the freely reduced word  $a := x^2yz^{-1}y$  and describes the automorphism  $f$  with the following regular Nielsen transformation

$$\begin{aligned} (x, y, z) &\xrightarrow{[(N2)_{1,2}]^2} (xy^2, y, z) \\ &\xrightarrow{(N2)_{3,2}} (xy^2, y, zy) \\ &\xrightarrow{(N1)_3} (xy^2, y, y^{-1}z^{-1}) \\ &\xrightarrow{(N2)_{2,3}} (xy^2, z^{-1}, y^{-1}z^{-1}); \end{aligned}$$

hence the automorphism is

$$\begin{aligned} f : F &\rightarrow F \\ x &\mapsto xy^2, \\ y &\mapsto z^{-1}, \\ z &\mapsto y^{-1}z^{-1} \end{aligned}$$

and she defines in GAP:

```
x1:=x*y^2;;
y1:=z^(-1);;
z1:=y^(-1)*z^(-1);;
```

Alice chooses as private key  $n = 7$ , hence she must calculate the automorphism  $f^7$ . For this she calculates in GAP:

```
#Calculate automorphism f^2=f^1(f^1)
x2:=x1*y1^2;;
y2:=z1^(-1);;
z2:=y1^(-1)*z1^(-1);;

gap> x2; y2; z2;
x*y^2*z^-2
z*y
z^2*y

#Calculate automorphism f^3=f^1(f^2)
x3:=x2*y2^2;;
y3:=z2^(-1);;
```

```

z3:=y2^(-1)*z2^(-1);;

gap> x3; y3; z3;
      x*y^2*z^-1*y*z*y
      y^-1*z^-2
      (y^-1*z^-1)^2*z^-1

#Calculate automorphism f^5=f^2(f^3)
x5:=x3*y3^2*z3^(-2);;
y5:=z3*y3;;
z5:=z3^2*y3;;

gap> x5; y5; z5;
      x*y^2*z^-1*y^2*z*(z*y)^2
      y^-1*(z^-1*y^-1*z^-1)^2*z^-1
      ((y^-1*z^-1)^2*z^-1)^2*y^-1*z^-2

#Calculate automorphism f^7=f^2(f^5)
x7:=x5*y5^2*z5^(-2);;
y7:=z5*y5;;
z7:=z5^2*y5;;

gap> x7; y7; z7;
      x*y^2*z^-1*y*(y*z)^2*(z*y*z^2*y)^2*z*y
      y^-1*((z^-1*y^-1*z^-1)^2*y^-1*z^-1)^2*z^-1*y^-1*z^-2
      (((y^-1*z^-1)^2*z^-1)^2*y^-1*z^-2)^2*y^-1*(z^-1*y^-1*z^-1)^2*z^-1

```

Thus, the automorphism  $f^7$  is

$$\begin{aligned}
 f^7 : F &\rightarrow F \\
 x &\mapsto xy^2z^{-1}y(yz)^2(zyz^2y)^2zy, \\
 y &\mapsto y^{-1}((z^{-1}y^{-1}z^{-1})^2y^{-1}z^{-1})^2z^{-1}y^{-1}z^{-2}, \\
 z &\mapsto (((y^{-1}z^{-1})^2z^{-1})^2y^{-1}z^{-2})^2y^{-1}(z^{-1}y^{-1}z^{-1})^2z^{-1}.
 \end{aligned}$$

Her public key is  $c := f^7(a)$ :

```

c:=x7^2*y7*z7^(-2)*y7;;

gap> c;
      (x*y^2*z^-1*y*(y*z)^2*(z*y*z^2*y)^2*z*y)^2*(z^2*y)^2*\
      ((z*y*z)^2*y*z)^2*z*y*z^2*y*z^-1

```

Bob is now able to send a message to Alice. Let  $m = z^{-2}y^2zx^2y^{-1}x^{-1}$  be the message for Alice. He chooses the ephemeral key  $t = 5$  and hence calculates the automorphism  $f^5$  in GAP as follows:

```

m:=z^-2*y^2*z*x^2*y^-1*x^-1;;

```

```
#Calculate automorphism f^2=f^1(f^1)
x2:=x1*y1^2;;
y2:=z1^(-1);;
z2:=y1^(-1)*z1^(-1);;
```

```
gap> x2; y2; z2;
      x*y^2*z^-2
      z*y
      z^2*y
```

```
#Calculate automorphism f^3=f^1(f^2)
x3:=x2*y2^2;;
y3:=z2^(-1);;
z3:=y2^(-1)*z2^(-1);;
```

```
gap> x3; y3; z3;
      x*y^2*z^-1*y*z*y
      y^-1*z^-2
      (y^-1*z^-1)^2*z^-1
```

```
#Calculate automorphism f^5=f^2(f^3)
x5:=x3*y3^2*z3^(-2);;
y5:=z3*y3;;
z5:=z3^2*y3;;
```

```
gap> x5; y5; z5;
      x*y^2*z^-1*y^2*z*(z*y)^2
      y^-1*(z^-1*y^-1*z^-1)^2*z^-1
      ((y^-1*z^-1)^2*z^-1)^2*y^-1*z^-2
```

Hence, the automorphism  $f^5$  is

$$\begin{aligned}
 f^5 : F &\rightarrow F \\
 x &\mapsto xy^2z^{-1}y^2z(zy)^2, \\
 y &\mapsto y^{-1}(z^{-1}y^{-1}z^{-1})^2z^{-1}, \\
 z &\mapsto ((y^{-1}z^{-1})^2z^{-1})^2y^{-1}z^{-2}.
 \end{aligned}$$

He now calculates his ciphertext  $(c_1, c_2)$  for Alice with  $c_1 = m \cdot f^5(c)$  and  $c_2 = f^5(a)$  in GAP:

```
#c22:=f^5(c)
c22:=(x5*y5^2*z5^(-1)*y5*(y5*z5)^2*(z5*y5*z5^2*y5)^2*z5*y5)^2*\
      (z5^2*y5)^2*((z5*y5*z5)^2*y5*z5)^2*z5*y5*z5^2*y5*z5^(-1);;
c1:=m*c22;;
```

```
gap> c1;
      z^-2*y^2*z*x^2*(y*z^-1)^2*((z^-1*y^-1*z^-2*y^-1)^2\
      *z^-2*y^-1)^2*(z^-1*y^-1*z^-1)^2*z^-1*y^-1*(((z^-1\
```

```

1*y^-1*z^-1)^2*y^-1*z^-1)^2*z^-1*y^-1*z^-1*y^-1*z^-1\
-1)^2*(z^-1*y^-1*z^-2*y^-1)^2*z^-1*y^-1*z^-1)^2*((\
z^-1*y^-1*z^-2*y^-1)^2*z^-2*y^-1)^2*(z^-1*y^-1*z^-1\
1)^2*z^-1*x*y^2*z^-1*y*(z^-1*((z^-1*y^-1*z^-2*y^-1\
1)^2*z^-2*y^-1)^2*(z^-1*y^-1*z^-1)^2*z^-1*y^-1)^3*\
(z^-1*y^-1*z^-1)^2*y^-1*z^-1*((z^-1*y^-1*z^-2*y^-1\
)^2*z^-2*y^-1)^2*(z^-1*y^-1*z^-1)^2*y^-1)^3*z^-1*(\
(z^-1*y^-1*z^-2*y^-1)^2*z^-2*y^-1)^2*(z^-1*y^-1*z^-1\
-1)^2*y^-1*z^-1*y

```

```

#c2:=f^5(a)
c2:=x5^2*y5*z5^(-2)*y5;;

```

```

gap> c2;
(x*y^2*z^-1*y^2*z*(z*y)^2)^2*z^2*y*(z*y*z)^2*z*y*z^-1

```

Bob sends  $(c_1, c_2)$  to Alice. Alice gets the message  $m$  by calculating

$$m = c_1 \cdot (f^7(c_2))^{-1}.$$

In GAP she computes:

```

#dc:=f^7(c2)
dc:=(x7*y7^2*z7^(-1)*y7^2*z7*(z7*y7)^2)^2*z7^2*y7*\
(z7*y7*z7)^2*z7*y7*z7^(-1);;

```

```

gap> dc;
(x*y*(y*z^-1)^2*((z^-1*y^-1*z^-2*y^-1)^2*z^-2*y^-1\
)^2*(z^-1*y^-1*z^-1)^2*z^-1*y^-1*((((z^-1*y^-1*z^-1\
1)^2*y^-1*z^-1)^2*z^-1*y^-1*z^-1*y^-1*z^-1)^2*(z^-1\
1*y^-1*z^-2*y^-1)^2*z^-1*y^-1*z^-1)^2*((z^-1*y^-1*\
z^-2*y^-1)^2*z^-2*y^-1)^2*(z^-1*y^-1*z^-1)^2*z^-1)\
^2*y^-1*((z^-1*y^-1*z^-1)^2*y^-1*z^-1)^2*z^-1*y^-1\
1*z^-1*y^-1*z^-1)^2*(z^-1*y^-1*z^-2*y^-1)^2*z^-1*y\
^-1*z^-1*((((z^-1*y^-1*z^-2*y^-1)^2*z^-2*y^-1)^2*(\
z^-1*y^-1*z^-1)^2*z^-1*y^-1)^2*(z^-1*y^-1*z^-1)^2\
*y^-1*z^-1)^2*z^-1*y^-1*z^-2*y^-1)^2*((z^-1*y^-1*\
z^-1)^2*y^-1*z^-1)^2*z^-1*y^-1*z^-1*y^-1*z^-1)^2*(\
z^-1*y^-1*z^-2*y^-1)^2*z^-1*y^-1*z^-1*y

```

```

gap> dc^-1;
y^-1*(((z*y)^2*z)^2*z*y*z)^2*z*y*(z*y*z)^2*z*\
y*((z*y*z)^2*y*z)^2*z*y*z)^2*(z*y*((z*y*z)^2*y*z)\
^2*z*y*z*y*z)^2*(z*y*z^2*y)^2*z)^2*y*((((z^2*y)^2\
*z*y)^2*z^2*y*z*y)^2*z*(z*y*z^2*y)^2*z*y)^2*z*(z*y\
*z^2*y)^2*z^2*y*((z*y*z)^2*y*z)^2*z*y*z*y*z)^2*(z\
*y*z^2*y)^2*z*(z*y^-1)^2*y^-1*x^-1)^2

```

```
gap> c1*dc^-1;  
z^-2*y^2*z*x^2*y^-1*x^-1
```

Finally, she reconstructs the correct message

$$z^{-2}y^2zx^2y^{-1}x^{-1}.$$

## C.12. Example 10.2.2 executed with GAP and Maple 16

In a challenge and response system, the prover and the verifier agree privately on a password and a challenge automorphisms  $f$ . In this example they agree on the password  $P = \text{Bob}$  and the challenge automorphism  $f$  is described with the regular Nielsen transformation

$$\begin{aligned} (x, y, z) &\xrightarrow{[(N2)_{1,2}]^2} (xy^2, y, z) \\ &\xrightarrow{(N2)_{3,2}} (xy^2, y, zy) \\ &\xrightarrow{(N1)_3} (xy^2, y, y^{-1}z^{-1}) \\ &\xrightarrow{(N2)_{2,3}} (xy^2, z^{-1}, y^{-1}z^{-1}); \end{aligned}$$

hence the automorphism is

$$\begin{aligned} f : F &\rightarrow F \\ x &\mapsto xy^2, \\ y &\mapsto z^{-1}, \\ z &\mapsto y^{-1}z^{-1}. \end{aligned}$$

With the agreement on the automorphism  $f$ , it is known, that  $X = \{x, y, z\}$ . The prover should verify himself to the verifier. For this, he sends the password  $P = \text{Bob}$  to the verifier. Now, the verifier knows the corresponding challenge automorphism  $f$  to the password  $P$  and generates a challenge for the prover. In this example, he chooses the natural number  $n = 7$ , the reduced word  $w = x^2y^2z^{-1}$  and the following faithful representation

$$\begin{aligned} \varphi : F &\rightarrow \text{SL}(2, \mathbb{Q}) \\ x &\mapsto \begin{pmatrix} -409 & 1394 \\ 120 & -409 \end{pmatrix}, \\ y &\mapsto \begin{pmatrix} \frac{435}{2} & \frac{-6479}{4} \\ -19 & \frac{283}{2} \end{pmatrix}, \\ z &\mapsto \begin{pmatrix} \frac{16843}{2} & \frac{-113025}{4} \\ -1133 & \frac{7603}{2} \end{pmatrix}. \end{aligned}$$

The matrices  $\varphi(x)$ ,  $\varphi(y)$  and  $\varphi(z)$  generate a free group, because they were generated with Theorem 4.2.18 and a Nielsen reduced set in  $G = \langle a, b, c \mid \rangle$  as follows:

The verifier uses Maple 16 and chooses the following rational elements  $r[1]$ ,  $r[2]$  and  $r[3]$  and proves the inequalities (4.1) from Theorem 4.2.18:

```
> restart; with(LinearAlgebra):
> r[1] := 7/2;
> r[2] := 15/2;
> r[3] := 23/2;
```

$$\begin{aligned} r_1 &:= \frac{7}{2} \\ r_2 &:= \frac{15}{2} \\ r_3 &:= \frac{23}{2} \end{aligned}$$

```
> r[1]-2;
> r[2]-r[1]-3;
> r[3]-r[2]-3;
```

$$\frac{3}{2}$$

$$1$$

$$1$$

All results are greater than 0, hence he can generate with the numbers  $r[1]$ ,  $r[2]$  and  $r[3]$  matrices which generate a free subgroup of  $SL(2, \mathbb{Q})$  of rank 3. The verifier then gets with Theorem 4.2.18 the following matrices  $M[1]$ ,  $M[2]$  and  $M[3]$ , which generate a free group:

```
> M[1] := Matrix([[ -r[1], r[1]^2-1], [1, -r[1]]]);
> M[2] := Matrix([[ -r[2], r[2]^2-1], [1, -r[2]]]);
> M[3] := Matrix([[ -r[3], r[3]^2-1], [1, -r[3]]]);
```

$$M_1 := \begin{bmatrix} -\frac{7}{2} & \frac{45}{4} \\ 1 & -\frac{7}{2} \end{bmatrix}$$

$$M_2 := \begin{bmatrix} -\frac{15}{2} & \frac{221}{4} \\ 1 & -\frac{15}{2} \end{bmatrix}$$

$$M_3 := \begin{bmatrix} -\frac{23}{2} & \frac{525}{4} \\ 1 & -\frac{23}{2} \end{bmatrix}$$

Due to the fact, that these matrices have a special look, the verifier uses a Nielsen reduced set  $U$  with 3 elements in a free group  $G = \langle a, b, c \mid \ \rangle$  to generate a free subgroup of  $G$ . In GAP it is:

```
LoadPackage("FGA");;
G:=FreeGroup("a", "b", "c");;
AssignGeneratorVariables(G);;

GG:=Group(a*b*a, c*b, b^2*a^2);;

gap> FreeGeneratorsOfGroup(GG);
[ a*b*a, c*b, b^2*a^2 ]
```

With the operation

```
▷ FreeGeneratorsOfGroup(GG)
```

which returns a list of free Nielsen reduced generators, which defines a Nielsen reduced set of the finitely generated subgroup  $GG$  of a free group. Thus, the verifier proved that

$$U = \{aba, cb, b^2a^2\}$$

is Nielsen reduced.

Therefore, with  $a := M[1]$ ,  $b := M[2]$  and  $c := M[3]$  he generates the elements  $\varphi(x) = \varphi(aba) =: x1$ ,  $\varphi(y) = \varphi(cb) =: y1$  and  $\varphi(z) = \varphi(b^2a^2) =: z1$ , thus it is:



```

> x1:=M[1].M[2].M[1];
> y1:=M[3].M[2];
> z1:=M[2].M[2].M[1].M[1];

```

$$x1 := \begin{bmatrix} -409 & 1394 \\ 120 & -409 \end{bmatrix}$$

$$y1 := \begin{bmatrix} \frac{435}{2} & \frac{-6479}{4} \\ -19 & \frac{283}{2} \end{bmatrix}$$

$$z1 := \begin{bmatrix} \frac{16843}{2} & \frac{-113025}{4} \\ -1133 & \frac{7603}{2} \end{bmatrix}$$

Hence, the above faithful representation  $\varphi$  is generated.

The verifier sends  $(\varphi, w, n)$  to the prover, with  $n = 7$  and  $w = x^2y^2z^{-1}$ .

The prover and the verifier perform now the same steps. They calculate  $\varphi(f^n(w))$ .

First, they calculate the automorphism  $f^7$ ; for this they use GAP as follows:

```
#Calculate automorphism f^2=f^1(f^1)
```

```
x2:=x1*y1^2;;
```

```
y2:=z1^(-1);;
```

```
z2:=y1^(-1)*z1^(-1);;
```

```
gap> x2; y2; z2;
```

```
  x*y^2*z^-2
```

```
  z*y
```

```
  z^2*y
```

```
#Calculate automorphism f^3=f^1(f^2)
```

```
x3:=x2*y2^2;;
```

```
y3:=z2^(-1);;
```

```
z3:=y2^(-1)*z2^(-1);;
```

```
gap> x3; y3; z3;
```

```
  x*y^2*z^-1*y*z*y
```

```
  y^-1*z^-2
```

```
  (y^-1*z^-1)^2*z^-1
```

```
#Calculate automorphism f^5=f^2(f^3)
```

```
x5:=x3*y3^2*z3^(-2);;
```

```
y5:=z3*y3;;
```

```
z5:=z3^2*y3;;
```

```
gap> x5; y5; z5;
```

```
  x*y^2*z^-1*y^2*z*(z*y)^2
```

```
  y^-1*(z^-1*y^-1*z^-1)^2*z^-1
```

```
  ((y^-1*z^-1)^2*z^-1)^2*y^-1*z^-2
```

```
#Calculate automorphism f^7=f^2(f^5)
x7:=x5*y5^(2)*z5^(-2);;
y7:=z5*y5;;
z7:=z5^2*y5;;

gap> x7; y7; z7;
x*y^2*z^-1*y*(y*z)^2*(z*y*z^2*y)^2*z*y
y^-1*((z^-1*y^-1*z^-1)^2*y^-1*z^-1)^2*z^-1*y^-1*z^-2
(((y^-1*z^-1)^2*z^-1)^2*y^-1*z^-2)^2*y^-1*(z^-1*y^-1*z^-1)^2*z^-1
```

Therefore, the automorphism  $f^7$  is

$$f^7 : F \rightarrow F$$

$$x \mapsto xy^2z^{-1}y(yz)^2(zyz^2y)^2zy,$$

$$y \mapsto y^{-1}((z^{-1}y^{-1}z^{-1})^2y^{-1}z^{-1})^2z^{-1}y^{-1}z^{-2},$$

$$z \mapsto (((y^{-1}z^{-1})^2z^{-1})^2y^{-1}z^{-2})^2y^{-1}(z^{-1}y^{-1}z^{-1})^2z^{-1}.$$

They need the element  $r := f^7(x^2y^2z^{-1})$ :

```
r:=x7^2*y7^2*z7^-1;;

gap> r;
x*y^2*z^-1*y*(y*z)^2*(z*y*z^2*y)^2*z*y*x*y^2*z^-1*y^2*(z*y*z)^2*y
```

With the element  $r$  they can now calculate the matrix  $M = \varphi(f^7(x^2y^2z^{-1})) = \varphi(r)$ . Thus, they use Maple 16 with the matrices  $x1$ ,  $x2$  and  $x3$ :

```
> x1:=M[1].M[2].M[1];
> y1:=M[3].M[2];
> z1:=M[2].M[2].M[1].M[1];
```

$$x1 := \begin{bmatrix} -409 & 1394 \\ 120 & -409 \end{bmatrix}$$

$$y1 := \begin{bmatrix} \frac{435}{2} & \frac{-6479}{4} \\ -19 & \frac{283}{2} \end{bmatrix}$$

$$z1 := \begin{bmatrix} \frac{16843}{2} & \frac{-113025}{4} \\ -1133 & \frac{7603}{2} \end{bmatrix}$$

Now,  $M$  can be calculated as follows:

```
> M:=x1.y1.y1.MatrixInverse(z1).y1.y1.z1.y1.z1.z1.y1.z1.z1.y1.z1.y1.z1.z1.
> y1.z1.y1.x1.y1.y1.MatrixInverse(z1).y1.y1.z1.y1.z1.z1.y1.z1.y1;
```

```

M := [23841548704365461063218823154411079276122356828819682877543\
05517896347784578880505953540504181284250715226083, -17755233\
53499978017039773651578189514853519664889709072698793500963\
8424849031800573503371171870460697578430073]
[-6995085008480279366250726044105857770410774036773809665079\
93357004520123633034515140714187153570659417305932, 52093666\
17194712302044067796474373049405262712671263897905945954612\
344636304031828987283471508883904579622639]

```

Thus, the response is the matrix  $M = \begin{pmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{pmatrix}$  with

```

M1,1 = 2384154870436546106321882315441107927612235682881968287754305517896347784578880505953540504181284250715226083
M1,2 = -17755233534999780170397736515781895148535196648897090726987935009638424849031800573503371171870460697578430073
M2,1 = -699508500848027936625072604410585777041077403677380966507993357004520123633034515140714187153570659417305932
M2,2 = 5209366617194712302044067796474373049405262712671263897905945954612344636304031828987283471508883904579622639.

```

If the challenge is to give the last 10 digits of  $M_{2,2}$ , then the response is only 4579622639. The verifier compares the response with his result. If the response is correct, the prover is verified by the verifier.



# Bibliography

- [AFR05] P. Ackermann, B. Fine and G. Rosenberger, *On surface groups: Motivating examples in combinatorial group theory*, LMS Lecture Note Series **339**, Groups St. Andrews vol. 1 (2005), 126–170.
- [Atk89] K. Atkinson, *An introduction to numerical analysis*, 2nd edition, John Wiley & Sons, 1989.
- [BBFT10] G. Baumslag, Y. Bryiukov, B. Fine and D. Troeger, *Challenge response password security using combinatorial group theory*, De Gruyter Groups Complexity Cryptology **2** (2010), 67–81.
- [Bea83] A. F. Beardon, *The Geometry of Discrete Groups*, volume 91 of Graduate Texts in Mathematics, Springer Verlag, 1983.
- [BF08] G. Baumslag and B. Fine, *Augmented rings, matrices and public key cryptography: I*, private communication (notes from 2008).
- [BFKR15] G. Baumslag, B. Fine, M. Kreuzer and G. Rosenberger, *A course in mathematical cryptography*, De Gruyter, 2015.
- [BH93] B. Brink and R. B. Howlett, *A finiteness property and an automatic structure for Coxeter groups*, Mathematische Annalen **296** (1993), 179–190.
- [BL90] J. Benaloh and J. Leichter, *Generalized secret sharing and monotone functions*, CRYPTO '88 Proceedings on Advances in Cryptology, Springer Verlag New York (1990), 27–35.
- [Bla79] G. Blakley, *Safeguarding cryptographic keys*, Proceedings of the National Computer Conference **48** (1979), 313–317.
- [BMS02] A. V. Borovik, A. G. Myasnikov and V. Shpilrain, *Measuring sets in infinite groups*, Contemporary Mathematics **298** (2002), 21–42.
- [BNS10] A. Beutelspacher, H. B. Neumann and T. Schwarzpaul, *Kryptografie in Theorie und Praxis*, 2nd edition, Vieweg+Teubner, 2010.
- [Bos08] S. Bosch, *Lineare Algebra*, 4th edition, Springer Verlag, 2008.
- [BS65] H. Behnke and F. Sommer, *Theorie der analytischen Funktionen einer komplexen Veränderlichen*, 3rd edition, Springer Verlag, 1965.
- [Buc10] J. Buchmann, *Einführung in die Kryptographie*, Springer Verlag, 2010.
- [CFMRZ16] C. S. Chum, B. Fine, A. I. S. Moldenhauer, G. Rosenberger and X. Zhang, *On secret sharing protocols*, Contemporary Mathematics (to appear 2016).
- [CFRZ12] C. S. Chum, B. Fine, G. Rosenberger and X. Zhang, *A proposed alternative to the shamir secret sharing scheme*, Contemporary Mathematics **582** (2012), 47–50.

- [CgRR08] T. Camps, V. große Rebel and G. Rosenberger, *Einführung in die kombinatorische und die geometrische Gruppentheorie*, Berliner Studienreihe zur Mathematik Band 19, Heldermann Verlag, 2008.
- [CK02] P. Clote and E. Kranakis, *Boolean functions and computation models*, Springer Verlag, 2002.
- [Dav73] M. Davis, *Hilbert's Tenth Problem is Unsolvable*, The American Mathematical Monthly **80** (1973), no. 3, 233–269.
- [Deh11] M. Dehn, *Über unendliche diskontinuierliche Gruppen*, Mathematische Annalen **71** (1911), 116–144.
- [DH76] W. Diffie and M. E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **IT 22** (1976), 644–654.
- [DKR13] V. Diekert, M. Kufleitner and G. Rosenberger, *Diskrete algebraische Methoden*, De Gruyter, 2013.
- [EKLG14] B. Eick, M. Kirschmer and C. Leedham-Green, *The constructive membership problem for discrete free subgroups of rank 2 of  $SL_2(\mathbb{R})$* , LMS Journal of Computation and Mathematics **17** (2014), 345–359.
- [ElG85] T. ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory **IT-31** (1985), 469–473.
- [EMNW11] G. Engeln-Müllges, K. Niederdrenk and R. Wodicka, *Numerik-Algorithmen, Verfahren, Beispiele, Anwendungen*, 10th revised and extended edition, Springer Verlag, 2011.
- [FGMRS14] B. Fine, A. Gaglione, A. Myasnikov, G. Rosenberger and D. Spellman, *The Elementary Theory of Groups*, De Gruyter, 2014.
- [FHKR11] B. Fine, M. Habeeb, D. Kahrobaei and G. Rosenberger, *Aspects of nonabelian group based cryptography: A survey and open problems*, JP Journal of Algebra, Number Theorie and Applications **21** (2011), 1–40.
- [FKIMR15] B. Fine, G. Kern-Isberner, A. I. S. Moldenhauer and G. Rosenberger, *On the Generalized Hurwitz Equation and the Baragar-Umeda Equation*, Results in Mathematics **69** (2015), 69–92.
- [FKR14] B. Fine, M. Kreuzer and G. Rosenberger, *Faithful real representation of cyclically pinched one-relator groups*, International Journal of Group Theory **3** (2014), no. 1, 1–8.
- [FMR13] B. Fine, A. I. S. Moldenhauer and G. Rosenberger, *A secret sharing scheme based on the Closest Vector Theorem and a modification to a private key cryptosystem*, De Gruyter Groups Complexity Cryptology **5** (2013), 223–238.
- [FR99] B. Fine and G. Rosenberger, *Algebraic Generalizations of Discrete Groups: A Path to Combinatorial Group Theory through One-Relator Products*, CRC Press, 1999.
- [Fis10] G. Fischer, *Lineare Algebra*, 17th edition, Vieweg+Teubner Verlag, 2010.
- [GAP15] GAP, *Version 4.7.7 of 13-feb-2015 (free software, GPL)*, <http://www.gap-system.org> (2015).

- [GS07] Y. Gurevich and P. Schupp, *Membership problem for the modular group*, *Slam J. Comput.* **37** (2007), no. 2, 425–459.
- [Hil02] D. Hilbert, *Mathematical problems*, *Bulletin of the American Mathematical Society* **8** (1902), 437–479.
- [HKKS13] M. Habeeb, D. Kahrobaei, C. Koupparis and V. Shpilrain, *Public key exchange using semidirect product of (semi)groups*, *ACNS 2013, Lecture Notes in Computer Science* **7954** (2013), 475–486.
- [HPS08] J. Hoffstein, J. Pipher and J. H. Silverman, *An introduction to mathematical cryptography*, Springer Verlag, 2008.
- [ISN87] M. Ito, A. Saito and T. Nishizeki, *Secret sharing scheme realizing general access structure*, *IEEE Global Telecommunications Conference* (1987), 99–102.
- [ISN93] M. Ito, A. Saito and T. Nishizeki, *Multiple assignment scheme for sharing secret*, *Journal of Cryptology* (1993), 15–20.
- [Jan70] G. J. Janusz, *Faithful representation of  $p$ -groups at characteristic  $p$* , *Journal of Algebra* **15** (1970), 335–351.
- [JS06] J. C. Jantzen and J. Schwermer, *Algebra*, Springer Verlag, 2006.
- [Kah96] D. Kahn, *The codebreakers: The comprehensive history of secret communication from ancient times to the internet*, Scribner, 1996.
- [KK06] D. Kahrobaei and B. Khan, *A non-commutative generalization of ElGamal key exchange using polycyclic groups*, *Proceeding of IEEE, GLOBECOM* (2006), 1–5.
- [KK12] D. Kahrobaei and C. Koupparis, *Non-commutative digital signatures*, *De Gruyter Groups Complexity Cryptology* **4** (2012), 377–384.
- [KKS13] D. Kahrobaei, C. Koupparis and V. Shpilrain, *Public key exchange using matrices over group rings*, *De Gruyter Groups Complexity Cryptology* **5** (2013), 97–115.
- [KLS15] D. Kahrobaei, H. T. Lam and V. Shpilrain, *Public key exchange using extensions by endomorphisms and matrices over a galois field*, preprint <http://www.sci.ccnycuny.edu/~shpil/res.html>, 2015.
- [KMU14] M. Kreuzer, A. D. Myasnikov and A. Ushakov, *A linear algebra attack to group-ring-based key exchange protocols*, *Applied Cryptography and Network Security* vol. 8479 of the series *Lecture Notes in Computer Science* (2014), 37–43.
- [Kna92] A. W. Kna, *Elliptic curves*, Princeton University Press, 1992.
- [Kob87] N. Koblitz, *A course in number theory and cryptography*, Springer Verlag, 1987.
- [KS16] D. Kahrobaei and V. Shpilrain, *Using semidirect product of (semi)groups in public key cryptography*, *ArXiv: <http://arxiv.org/abs/1604.05542>* (2016).
- [Leh64] J. Lehner, *Discontinuous groups and automorphic functions*, *Mathematical Surveys Number VIII*, American Mathematical Society, Providence, Rhode Island, 1964.
- [Lig06] A. Ligêza, *Lokal foundations for rule-based systems*, vol. 11, Springer Verlag, 2006.

- [Liu68] C. L. Liu, *Introduction to combinatorial mathematics*, McGraw-Hill, 1968.
- [LS77] R. C. Lyndon and P. E. Schupp, *Combinatorial group theory*, Ergebnisse der Mathematik und ihre Grenzgebiete **89**, Springer Verlag, 1977.
- [Mag73] W. Magnus, *Rational Representations of Fuchsian Groups and Non-parabolic Subgroups of the Modular Group*, Nachrichten der Akademie der Wissenschaft in Göttingen (1973), 179–189.
- [Mat70] Y. Matiyasevich, *Solution of the Tenth Problem of Hilbert*, Mat. Lapok **21** (1970), 83–87.
- [Mat96] Y. Matiyasevich, *Hilbert's Tenth Problem: What can we do with Diophantine equations?*, <http://logic.pdmi.ras.ru/~yumat/personaljournal/reversechronocontent.html>, 1996.
- [MKS66] W. Magnus, A. Karrass and D. Solitar, *Combinatorial group theory*, Pure and Applied Mathematics, A Series of Texts and Monographs Volume XIII, John Wiley & Sons, 1966.
- [Mol12] A. I. S. Moldenhauer, *Untersuchungen der Secret-Sharing-Protokolle von Shamir und Panagopoulos, sowie die Entwicklung eines neuen Secret-Sharing-Protokolls*, Master's Thesis, University of Hamburg, 2012.
- [Mol15] A. I. S. Moldenhauer, *A group theoretical ElGamal cryptosystem based on a semidirect product of groups and a proposal for a signature protocol*, Contemporary Mathematics **633** (2015), 97–113.
- [MR15] A. I. S. Moldenhauer and G. Rosenberger, *Cryptographic protocols based on Nielsen transformations*, ArXiv: <https://arxiv.org/abs/1504.03141v1> (2015).
- [MR16] A. I. S. Moldenhauer and G. Rosenberger, *Cryptosystems using automorphisms of finitely generated free groups*, Tributes **29**, Computational Models of Rationality (2016), 31–51.
- [MS03] A. G. Myasnikov and V. Shpilrain, *Automorphic orbits in free groups*, Journal of Algebra **269** (2003), 18–27.
- [MSU08] A. Myasnikov, V. Shpilrain and A. Ushakov, *Group-based cryptography*, Advanced Courses in Mathematics - CRM Barcelona, Birkhäuser Basel, 2008.
- [MvOV97] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of applied cryptography*, CRC Press LLC, 1997.
- [MW85] M. R. Magyarik and N. R. Wagner, *A public key cryptosystem based on the word problem*, Advances in Cryptology - CRYPTO '84, Lecture Notes in Computer Science **196**, Springer Verlag (1985), 19–36.
- [MW97] A. J. Menezes and Y.-H. Wu, *The discrete logarithm problem in  $GL(n, q)$* , Ars Combinatoria **47** (1997), 23–32.
- [Pan10] D. Panagopoulos, *A secret sharing scheme using groups*, ArXiv: <http://arxiv.org/abs/1009.0026> (2010).
- [PH78] S. C. Pohlig and M. E. Hellman, *An improved algorithm for computing logarithms over  $GF(p)$  and its cryptographic significance*, IEEE Transactions on Information Theory **24** (1978), 106–110.



- 
- [Rom13a] V. A. Roman'kov, *Algebraic cryptography*, Omsk State Dostoevsky University (2013).
- [Rom13b] V. A. Roman'kov, *Cryptanalysis of some schemes applying automorphisms*, *Prikladnaya Discretnaya Matematika* **3** (2013), 35–51.
- [Rom15] V. A. Roman'kov, *Linear decomposition attack on public key exchange protocols using semidirect products of (semi)groups*, ArXiv: <https://arxiv.org/abs/1501.01152v1> (2015).
- [Rot95] J. J. Rotman, *An introduction to the theory of groups*, Springer Verlag, 1995.
- [RSA78] R. L. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, *Commun. ACM* **21** (1978), no. 2, 120–126.
- [Sac96] V. N. Sachkov, *Combinatorial methods in discrete mathematics*, Cambridge University Press, 1996.
- [Sha79] A. Shamir, *How to share a secret*, *Communications of the ACM* **22** (1979), no. 11, 612–613.
- [Sho96] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, ArXiv: <https://arxiv.org/abs/quant-ph/9508027> (1996).
- [Sil09] J. H. Silverman, *The arithmetic of elliptic curves*, Springer Verlag, 2009.
- [Sin06] S. Singh, *Geheime Botschaften: Die Kunst der Verschlüsselung von der Antike bis in die Zeiten des Internets*, Deutscher Taschenbuch Verlag, 2006.
- [Spe82] E. Sperner, *Ein Satz über Untermengen einer endlichen Menge*, *Mathematische Zeitschrift* **27** (1982), 544–548.
- [Ste89] I. A. Stewart, *Obtaining Nielsen reduced sets in free groups*, Technical Report Series No. 293, 1989.
- [VS15] M. I. G. Vasco and R. Steinwandt, *Group theoretical cryptography*, CRC Press, 2015.
- [vTJ11] H. C. A. van Tilborg and S. Jajodia, *Encyclopedia of Cryptography and Security*, Springer Verlag, 2011.
- [Wat00] J. Watrous, *Succinct quantum proofs for properties of finite groups*, Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (2000), 537–546.
- [Weg87] I. Wegener, *The complexity of boolean functions*, Stuttgart: BG Teubner, 1987.



# Danksagung

Meinem Doktorvater Prof. Dr. Gerhard Rosenberger bin ich zutiefst dankbar für seine engagierte und hilfreiche Betreuung. Er hat mein Interesse an der gruppenbasierten Kryptology entfacht und durch das Masterstudium hindurch gefördert. Für die hervorragende Unterstützung während der Promotionsphase bin ich ihm sehr verbunden. Seine Tür stand mir stets offen und die Diskussionen waren immer sehr fruchtbar und spannend.

Ich möchte Prof. Dr. Ulf Kühn dafür danken, dass er mein Interesse an der mathematischen Kryptology schon im Bachelorstudium geweckt hat und für seine Unterstützung während der Promotionsphase.

Ich danke dem Team von Pro Exzellenzia für die finanzielle Unterstützung und deren hervorragendes Programm, das mich durch hilfreiche Workshops, Coachings und Netzwerkveranstaltungen während meiner Promotionsphase gestärkt und unterstützt hat.

Nicole Beisiegel, Mirjam Braßler, Veronika Altpeter und Britta Moldenhauer danke ich für ihr sorgfältiges Lesen und ihren Anmerkungen zu meiner Arbeit.

Markus Nikolaus danke ich für seine Geduld und sein Verständnis während dieser besonderen Zeit.

Meiner Familie bin ich dankbar für die Unterstützung während meines gesamten Studiums. Insbesondere möchte ich meinen Eltern für ihr Verständnis, ihre Ruhe und ihr offenes Ohr danken.

# Zusammenfassung

Das Thema dieser Arbeit ist angesiedelt in dem Gebiet der mathematischen Kryptologie, insbesondere in der gruppenbasierenden Kryptologie. Wir erweitern bestehende kryptographische Protokolle, entwickeln neue kryptographische Protokolle bezüglich des mathematischen Hintergrundes und geben Modifikationen für diese an. Erweitert wird die Arbeit durch Kryptoanalysen und Beispiele. Der Schwerpunkt dieser Arbeit liegt auf der Entwicklung von neuen kryptographischen Protokollen basierend auf nichtkommutativen Gruppen und Techniken, die typischerweise in der kombinatorischen Gruppentheorie Anwendung finden. Es werden Automorphismen von endlich erzeugten freien Gruppen genutzt, die mit Hilfe von Nielsen-Transformationen oder Whitehead-Automorphismen erzeugt werden können. Mit der Hilfe von Whitehead-Automorphismen entwickeln wir einen Vorschlag, um zufällig Automorphismen der Automorphismengruppe  $Aut(F)$  zu erzeugen, hierbei ist  $F$  eine endlich erzeugte freie Gruppe.

Es werden insgesamt zwölf kryptographische Protokolle vorgestellt, die für diese Arbeit entwickelt wurden. Darunter sind zwei Erweiterungen eines  $(n, t)$ -Secret-Sharing-Verfahrens, das auf einer Idee von C. S. Chum, B. Fine, G. Rosenberger und X. Zhang basiert. Sie beruhen auf dem Dichtesten-Vektor-Theorem in einem euklidischen Vektorraum. Die erste Erweiterung (**Protokoll 1**) ist ein symmetrisches Kryptosystem. Die zweite ist ein Challenge-and-Response-Verfahren (**Protokoll 2**), das durch eine Variation auch als Zwei-Wege-Authentifizierung genutzt werden kann. Weiterhin werden zwei Erweiterungen des HKKS-Schlüsselaustausch-Protokolls von M. Habbeb, D. Kahrobaei, C. Koupparis und V. Shpilrain gegeben. Das HKKS-Schlüsselaustausch-Protokoll nutzt semidirekte Produkte von (Halb-)Gruppen und wird zu einem ElGamal ähnlichen asymmetrischen Kryptosystem (**Protokoll 3**) sowie zu einem Signatur-Protokoll (**Protokoll 4**) erweitert. Aktuell wird an dem HKKS-Schlüsselaustausch-Protokoll geforscht, so gibt es Angriffe basierend auf der linearen Algebra und es wird nach passenden Plattformen für dieses Verfahren gesucht. Diese Forschung betrifft auch das ElGamal ähnliche asymmetrische Kryptosystem und das Signatur-Protokoll. Ein kurzer Überblick über die Forschung zum HKKS-Schlüsselaustausch-Protokoll wird gegeben.

Darüber hinaus wird ein rein kombinatorisches Secret-Sharing-Verfahren (**Protokoll 5**) vorgestellt. Dieses nutzt die Verteilung der Geheimnisse, wie sie D. Panagopoulos für ein  $(n, t)$ -Secret-Sharing-Verfahren erklärt. Wir zeigen, dass diese Verteilung der Geheimnisse, nach D. Panagopoulos, ein Spezialfall eines Multiple-Assignment-Verfahrens ist, das von M. Ito, A. Saito und T. Nishizeki eingeführt wird. Des Weiteren wird gezeigt, dass das vorgestellte kombinatorische Secret-Sharing-Protokoll ähnlich zu einer Variation eines Secret-Sharing-Protokolls von J. Benaloh und J. Leichter ist. Die Idee der Erweiterung des kombinatorischen Verfahrens durch die Benutzung von Automorphismen von endlich erzeugten freien Gruppen führt zu zwei neuen Secret-Sharing-Verfahren. Ein Vergleich zu Shamirs Secret-Sharing-Verfahren wird gegeben. Das erste der beiden neuen Secret-Sharing-Verfahren, nutzt eine endlich erzeugte freie Gruppe  $F$ , eine endlich erzeugte freie Untergruppe in der  $SL(2, \mathbb{Q})$  und Nielsen-Transformationen (**Protokoll 6**). Es bildet die Basis für die **Protokolle 7-12**, die auch auf der kombinatorischen Gruppentheorie basieren. Das andere Secret-Sharing-Verfahren (**Protokoll 7**) benutzt eine endlich erzeugte freie Gruppe  $F = \langle X \mid \quad \rangle$ , eine Nielsen reduzierte Menge  $U \neq X$  und eine Nielsen äquivalente Menge  $V$  zu  $U$  und gibt somit den abschließenden Input für die neu entwickelten kryptographischen **Protokolle 8-12**, die das Hauptergebnis dieser Arbeit darstellen. Es war möglich zwei symmetrische Kryptosysteme mit ähnlichen Modifikationen (**Protokoll 8** und **Protokoll 9**), ein weiteres symmetrische Kryptosysteme (**Protokoll 10**),

ein ElGamal ähnliches asymmetrisches Kryptosystem (**Protokoll 11**) und ein Challenge-and-Response-Protokoll (**Protokoll 12**) neu zu entwickeln, die alle die kombinatorische Gruppentheorie und Automorphismen auf endlich erzeugten freien Gruppen nutzen. Je nach Protokoll beruht die Sicherheit auf einem linearen Kongruenzgenerator, dem diskreten Logarithmus-Problem für zyklische Untergruppen der Automorphismengruppe einer endlich erzeugten freien Gruppe, der unbekannt algorithmischen Lösung des (konstruktiven) Untergruppenzugehörigkeitsproblems in Matrixgruppen über rationalen Zahlen oder dem zehnten Problem von Hilbert.



# Abstract

The topic of this thesis is established in the area of mathematical cryptology, more precisely in group based cryptology. We give extensions of cryptographic protocols, develop new cryptographic protocols concerning the mathematical background and give modifications of them. In addition cryptographic analysis as well as examples are given. The focus lays on the development of new cryptographic protocols using non-commutative groups and of techniques, which are typically studied in combinatorial group theory. Automorphisms on finitely generated free groups are used, which can be generated by Nielsen transformations or Whitehead-Automorphisms. With the help of the Whitehead-Automorphisms we develop an approach for choosing automorphisms randomly of the automorphism group  $Aut(F)$ , with  $F$  a finitely generated free group. Altogether twelve cryptographic protocols are explained. Among these are two extensions of a  $(n, t)$ -secret sharing protocol, which is introduced by C. S. Chum, B. Fine, G. Rosenberger and X. Zhang. Both extensions depend on the Closest Vector Theorem in a real inner product space. The first one (**Protocol 1**) is a symmetric key cryptosystem and the second one is a challenge and response system (**Protocol 2**), which can be used by a variation as a two-way authentication. Furthermore, the HKKS-key exchange protocol by M. Habbeb, D. Kahrobaei, C. Koupparis and V. Shpilrain, which uses semidirect products of (semi)groups, is extended to an ElGamal like public key cryptosystem (**Protocol 3**) and to a signature protocol (**Protocol 4**). There is an ongoing research about the HKKS-key exchange protocol with linear algebra attacks as well as research about suitable platforms, which also affects the ElGamal like public key cryptosystem and the signature protocol. A short overview of the research is given in this thesis.

Furthermore, a purely combinatorial secret sharing scheme (**Protocol 5**) is introduced, which uses a share distribution method explained by D. Panagopoulos for a  $(n, t)$ -secret sharing scheme. We show that this share distribution method is a special case of a multiple assignment scheme introduced by M. Ito, A. Saito and T. Nishizeki. Furthermore, the introduced combinatorial secret sharing protocol is shown to be similar to a variation of a secret sharing protocol explained by J. Benaloh and J. Leichter. The idea of enhancing the combinatorial secret sharing scheme by using automorphisms on finitely generated free groups leads to two new secret sharing schemes. In addition a comparison to Shamir's secret sharing scheme is given. The first one is a secret sharing scheme using a finitely generated abstract free group  $F$ , a finitely generated free subgroup in  $SL(2, \mathbb{Q})$  and Nielsen transformations (**Protocol 6**). **Protocol 6** is the basis for **Protocol 7-12**, which are also based on combinatorial group theory. The other secret sharing scheme (**Protocol 7**) uses a finitely generated free group  $F = \langle X \mid \quad \rangle$ , a Nielsen reduced set  $U \neq X$  and a Nielsen equivalent set  $V$  to  $U$  and gives therefore the final input for the newly developed cryptographic **Protocols 8-12**, which are the main result in this thesis. Two new private key cryptosystems with similar modifications (**Protocol 8** and **Protocol 9**) were developed, another new private key cryptosystem (**Protocol 10**), a new ElGamal like public key cryptosystem (**Protocol 11**) and a new challenge and response system (**Protocol 12**), which all use the combinatorial group theory and automorphisms on finitely generated free groups. Depending on the protocols the security is based on a linear congruence generator, the discrete logarithm problem in cyclic subgroups of the automorphism group of a finitely generated free group, the unknown algorithmic solution of the (constructive) membership problem in matrix groups over the rational numbers or Hilbert's Tenth Problem.





# Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, 23.08.2016