Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Distributed Architectures for Data Pseudonymization and Anonymization in Medical Research

DISSERTATION

zur Erlangung des akademischen Grades Dr. rer. nat.
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik
der Universität Hamburg

vorgelegt von
Tom Petersen

Hamburg
September 2024

Assessors

Prof. Dr. Hannes Federrath

Prof. Dr. Florian Tschorsch

Further members of the examination commission

Prof. Dr. Janick Edinger

Prof. Dr. Jan Baumbach

Date of the oral defense

20.02.2025

Abstract

Medical research increasingly employs statistical methods for various purposes, such as identifying risk factors for diseases, advancing precision medicine, and evaluating treatment outcomes. These methods generally require extensive datasets derived from personal health information, which includes medical histories, lifestyle factors, treatment results, and even genetic data. But the growing reliance on this highly sensitive personal data raises serious privacy and security concerns. One way to balance these concerns with data demands is the application of data privacy techniques, such as pseudonymization and anonymization. These techniques enable the analysis of respective data while protecting individuals' privacy and reducing risks of re-identification. The distributed nature of medical research data presents significant challenges for the utilization of these measures, leading to the central topic of this thesis. This thesis explores the application of data privacy methods in distributed environments without central data collection and investigates the associated challenges, limitations, opportunities, and advantages.

The first part of this thesis reviews existing literature on data privacy measures, specifically pseudonymization, de-identification techniques, syntactic privacy models, and semantic privacy models. This includes technical aspects, including basic techniques and their strengths and weaknesses, as well as legal considerations, particularly the interpretation of relevant terms and concepts as well as the debate surrounding data privacy. The subsequent sections investigate the distributed application of data privacy techniques in selected scenarios representative for the domain of medical research. The first example involves the generation of pseudonyms in distributed environments where individuals contribute data at multiple data sources. Additionally to the generation of pseudonyms, in our next contribution, we provide a way to protect the disclosure of pseudonyms by distributing the process across multiple parties. The next example focuses on distributed anonymization protocols. Here, we identify weaknesses in an existing distributed syntactic privacy protocol and present an updated protocol version that addresses these weaknesses. The thesis concludes with a more practice-oriented contribution: a platform concept for privacy-preserving medical registries that allows for distributed data collection, which has been successfully utilized in real-world studies.

The intersection of data science, regulatory frameworks, and data privacy measures has significant implications to the future of medical research, and this thesis aims to contribute to the advancement of data privacy and security practices in this field.

Zusammenfassung

Der zunehmende Einsatz datenbasierter, statistischer Methoden in der medizinischen Forschung ermöglicht unter anderem die Identifikation von Risikofaktoren für Krankheiten, die Beurteilung von Behandlungsverfahren und Fortschritte in der personalisierten Medizin. Diese Methoden basieren im Allgemeinen auf großen Datenmengen, die aus personenbeziehbaren Gesundheitsdaten wie etwa der medizinischen Historie, Lebensgewohnheiten, Behandlungsergebnissen, und genetischen Profilen gewonnen werden. Die zunehmende Nutzung dieser hochsensiblen personenbezogenen Daten führt jedoch zu erheblichen Datenschutz- und Sicherheitsbedenken. Eine Möglichkeit zur Vermittlung zwischen diesen Bedenken und dem Datenbedarf der Forschung ist der Einsatz von Maßnahmen wie Pseudonymisierungs- und Anonymisierungstechniken. Diese Techniken ermöglichen die Analyse entsprechender Gesundheitsdaten, reduzieren jedoch das Risiko einer Re-Identifizierung und schützen so die Privatsphäre Betroffener. Die verteilte Natur medizinischer Forschungsdaten bringt jedoch wesentliche Herausforderungen für den Einsatz entsprechender Techniken in verteilten Umgebungen mit sich, was zum zentralen Problemfeld dieser Arbeit führt. Diese Arbeit erkundet die Anwendung von Datenschutzmaßnahmen in verteilten Umgebungen ohne zentrale Datensammlung und untersucht Herausforderungen, Einschränkungen und Vorteile ihres Einsatzes.

Im ersten Teil der Arbeit wird der aktuelle Forschungsstand zu Datenschutzmaßnahmen, insbesondere Pseudonymisierung, Deidentifizierungstechniken sowie syntaktischen und semantischen Datenschutzmodellen, dargestellt. Dies umfasst technische Aspekte, einschließlich grundlegender Techniken sowie ihrer Vor- und Nachteile, ebenso wie rechtliche Betrachtungen, insbesondere die Interpretation grundlegender Begriffe sowie die Debatte über die Bewertung von Datenschutztechniken. Die nachfolgenden Abschnitte untersuchen den Einsatz von Datenschutztechniken in verteilten Umgebungen anhand von repräsentativen Szenarien aus dem Bereich der medizinischen Forschung. Das erste Szenario befasst sich mit der Generierung von Pseudonymen in verteilten Umgebungen, in denen Individuen Daten in mehreren Datenquellen beisteuern können. Ergänzend zur Generierung von Pseudonymen wird im zweiten Forschungsbeitrag eine Möglichkeit für den Schutz des Aufdeckungsprozesses von Pseudonymen bereitgestellt, die auf der Verteilung des Prozesses auf mehrere Parteien besteht. Das nächste Beispiel fokussiert verteilte Anonymisierungsverfahren. Es werden Schwachstellen in einem verteilten Protokoll für syntaktische Privatsphäre identifiziert und es wird eine Version des Protokolls entworfen, die diese Schwachstellen verhindert. Die Arbeit schließt mit einem eher praxisorientierten Beitrag: einem Plattformkonzept für datenschutzfreundliche medizinische Register, das eine verteilte Datenerfassung ermöglicht und bereits erfolgreich in der Praxis eingesetzt wurde.

Die Schnittstelle zwischen Data Science, Regulierung und technischen Datenschutzmaßnahmen hat erhebliche Auswirkungen auf die Zukunft der medizinischen Forschung und diese Arbeit zielt darauf ab, zur Weiterentwicklung der Datenschutz- und Sicherheitspraktiken in diesem Bereich beizutragen.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Prof. Dr. Hannes Federrath, for awakening my interest in security and privacy-related topics. I am very thankful for the opportunity to do my PhD in his working group and for various pieces of advice – in professional and private regard. The trust he placed in my abilities and the freedom he afforded me throughout my journey have made me the researcher I am today. I am also deeply grateful to Prof. Dr. Mathias Fischer for his valuable comments over the years and Prof. Dr. Florian Tschorsch for his constructive feedback and suggestions that have significantly enhanced the quality of my work.

Next, I want to thank my colleagues from the working group *Security in Distributed Systems* for all of their support, for hours of lively discussions, and for celebrating as well as suffering together. In particular, thanks to Anne, for the best office-cooked coffee and motivating conversations on Monday mornings, Britta, for her helpfulness in all organizational matters, Christian, for nice collaborations and for providing his knowledge about data protection, Daniel, for introducing me to the magic of SMPC and for replacing magic with understanding, Ephraim, for supervising my bachelors and masters theses and winning me over to work in science, Erik, Janik, Jens L., Jens W., for introducing more green things to our offices, Johanna, for being the most sociable person in the working group – online and offline, Joshua, for all collaborations, work-related and private conversations, and mutual motivation – you have really earned the already awarded title *work spouse*, Kevin, Marius, Matthias, for the most interesting stories about... innovative use of technology as well as the best home-brewed beer in the office, Max, for collective hours of writing Elm code and for organizing fun workshops, Maya, Monina, for being a great first desk mate and for bringing movement in our workday, Niklas, Pascal, for the most interesting weekly Apple podcast I have ever listened to, Sadaf, Stefan, and Tobi, for hours of designing, developing, and deploying privacy-preserving applications together. Additionally, thanks to the members of our neighboring working group *Computer Networks* for a pleasant, collaborative environment and in particular to David, Dogan, Flo, for great conversations on the hallway and over the chess board, Malte, Steffen, and Tatjana.

Finally, I want to express my sincere gratitude to my entire family and friends for their support and encouragement throughout my academic journey. While there are too many of you to mention individually, I would like to acknowledge a few key people who have played a significant role in this journey. To my parents, whose belief in the value of education has motivated me to start this journey. I sincerely value their guidance, understanding, and help throughout my life. To my grandparents, for their honest interest, for inspiring stories and for always believing in my potential. To my parents-in-law for their support not only but especially since the birth of their grandson. To my wife, Giulia, for... everything... and for making sure that I take much-needed breaks, and my son, Matti, for teaching me the true priorities in life and the joy he brings into my world. I am truly grateful to have all of you by my side on this journey.

Contents

List of Figures xii

List of Tables xiv

Acronyms xv

1 Introduction 1
- 1.1 Research Questions and Methodology . . . 2
- 1.2 Contributions . . . 4
- 1.3 Publications and Real-World Use . . . 6
- 1.4 Structure . . . 6

2 Data Privacy Measures for Protecting Sensitive Data 8
- 2.1 Systematization of Disclosure Risks . . . 10
  - 2.1.1 Literature Overview . . . 10
  - 2.1.2 Unified Taxonomy . . . 12
- 2.2 Privacy-Utility Tradeoff . . . 15
- 2.3 Term Definitions . . . 18
  - 2.3.1 Personal Data and Personally Identifiable Information . . . 19
  - 2.3.2 Anonymization and De-Identification . . . 22
  - 2.3.3 Pseudonymization . . . 26
- 2.4 Pseudonymity . . . 28
  - 2.4.1 Scenarios and Responsible Parties . . . 28
  - 2.4.2 Pseudonymity and Linkability . . . 29
  - 2.4.3 Attacks on Pseudonymity . . . 30
  - 2.4.4 Techniques . . . 31
  - 2.4.5 Concluding Remarks . . . 33
- 2.5 De-Identification Techniques . . . 33
  - 2.5.1 Generalization . . . 34
  - 2.5.2 Suppression . . . 36
  - 2.5.3 Swapping . . . 36
  - 2.5.4 Sampling . . . 37
  - 2.5.5 Slicing and Anatomization . . . 37
  - 2.5.6 Perturbation . . . 38
  - 2.5.7 Post Randomization Method . . . 38
  - 2.5.8 Microaggregation . . . 38
  - 2.5.9 Synthetic Data . . . 39
  - 2.5.10 Concluding Remarks . . . 39
- 2.6 Syntactic Privacy Models . . . 40
  - 2.6.1 Attribute Types . . . 40
  - 2.6.2 k-Anonymity . . . 41
  - 2.6.3 l-Diversity . . . 43

2.6.4 t-Closeness . . . 44
2.6.5 Algorithms . . . 45
2.6.6 Further Privacy Models . . . 48
2.6.7 Curse of High Dimensionality . . . 49
2.6.8 Attacks on Syntactic Privacy Models . . . 50
2.6.9 Concluding Remarks . . . 56
2.7 Differential Privacy . . . 57
2.7.1 ε-Differential Privacy . . . 59
2.7.2 Properties of Differential Privacy . . . 60
2.7.3 Differentially Private Mechanisms . . . 61
2.7.4 (ε, δ)-Differential Privacy and the Gaussian Mechanism . . . 69
2.7.5 Choosing Parameters ε and δ . . . 70
2.7.6 Outlook on Further Topics . . . 76
2.7.7 Concluding Remarks . . . 79
2.8 Privacy Breaches . . . 80
2.8.1 Chicago Homicide Victims Dataset . . . 81
2.8.2 Massachusetts Health Dataset . . . 81
2.8.3 AOL Dataset . . . 81
2.8.4 Netflix Prize Dataset . . . 81
2.8.5 Social Networks Datasets . . . 82
2.8.6 Kaggle Social Network Challenge Dataset . . . 82
2.8.7 Mobile Phone Location Dataset . . . 82
2.8.8 Credit Card Dataset . . . 82
2.8.9 New York Taxi Dataset . . . 83
2.8.10 Australian Health Dataset . . . 83
2.8.11 edX Dataset . . . 83
2.8.12 Reconstruction Attack against Aggregated Statistics . . . 83
2.9 Subsumption of the Privacy Debate and Technical Measures . . . 84
2.9.1 Discussion Summary . . . 84
2.9.2 Technical Background for the Discussions . . . 86
2.9.3 Proposals for Policy Changes . . . 88

3 Distributed Global Pseudonyms **91**
3.1 Background . . . 92
3.1.1 Searchable Encryption . . . 92
3.1.2 Bilinear Pairings . . . 94
3.1.3 A Multi-Reader/Multi-Writer Searchable Encryption Scheme . . . 94
3.2 Scenario and Properties . . . 96
3.2.1 Scenario . . . 97
3.2.2 Properties . . . 98
3.3 Globally Consistent Pseudonyms via Searchable Encryption . . . 99
3.3.1 Solving the Key Distribution Problem . . . 100
3.3.2 Extending the Scheme with Multi-Keyword Search . . . 101
3.3.3 Extending the Scheme with Fuzzy Search Capabilities . . . 103
3.3.4 Limiting the Linkability . . . 105
3.4 Adversary Model . . . 106
3.5 Implementation and Performance Evaluation . . . 107

3.6 Related Work . . . 110
3.6.1 PEEPLL . . . 110
3.6.2 Mainzelliste . . . 111
3.6.3 Optimized Mainzelliste based on PPRL . . . 111
3.6.4 MainSEL . . . 112
3.6.5 ScrambleDB . . . 113
3.6.6 Comparison to our Work . . . 114
3.7 Conclusion . . . 114

4 Pseudonym Disclosure based on the Multi-Eye-Principle 116
4.1 Background . . . 118
4.1.1 Shamir's Secret Sharing . . . 118
4.1.2 Threshold Decryption . . . 119
4.1.3 A Threshold Decryption Scheme . . . 119
4.1.4 Distributed Key Generation . . . 120
4.1.5 Proxy Re-Encryption . . . 121
4.1.6 A Bidirectional Multi-Hop Proxy Re-Encryption Scheme . . . 122
4.2 Pseudonym Disclosure Protected by Multi-Eye Principle . . . 123
4.3 Situations and Properties . . . 124
4.4 Existing Approaches for Access Structure Updates . . . 125
4.4.1 Naïve solution . . . 125
4.4.2 Monotonous access structure updates . . . 126
4.4.3 Dynamic secret sharing . . . 126
4.4.4 Quorum-controlled approach . . . 127
4.4.5 Blinding-based approach . . . 127
4.4.6 Summary . . . 127
4.5 Employing Proxy Re-Encryption for Access Structure Updates . . . 127
4.5.1 Scheme Instantiation . . . 130
4.5.2 Proof of Correctness . . . 131
4.5.3 Security Arguments . . . 133
4.5.4 Extension to Malicious Adversaries . . . 133
4.5.5 Applicability to Situations . . . 134
4.5.6 Properties of the Scheme . . . 135
4.6 Adversary Model for Pseudonym Disclosure . . . 136
4.7 Implementation and Evaluation . . . 137
4.8 Conclusion . . . 142

5 Distributed Syntactic Privacy 144
5.1 Background . . . 145
5.1.1 Distributed Syntactic Privacy . . . 145
5.1.2 Secure Multi-Party Computation . . . 145
5.2 A Vulnerable Distributed Syntactic Privacy Protocol . . . 147
5.2.1 Protocol Description . . . 148
5.2.2 Privacy and Security Claims . . . 150
5.2.3 Weaknesses in the Protocol . . . 150
5.2.4 Adversary Model of the Original Protocol . . . 151
5.3 Improving the Protocol through Secure Multi-Party Computation . . . 152
5.3.1 Basic Idea . . . 152

5.3.2 Implementation via SMPC 153
5.3.3 Updated Adversary Model 155
5.4 Performance Evaluation 156
5.5 Conclusion 160

6 A Privacy-Preserving Medical Registry Platform 161
6.1 Functional Description 162
6.1.1 Role Model 164
6.1.2 States of Data Records 165
6.1.3 Monitoring 165
6.1.4 Benchmarking 166
6.1.5 Offering Data Subject Rights 166
6.2 Security Architecture and Measures 167
6.2.1 Personal and Medical Data 167
6.2.2 Adversary Model 169
6.2.3 Cryptographically Enforced Client Separation and Pseudonymization170
6.2.4 User Authentication 171
6.2.5 Strong Passwords and Second Factor Authentication 173
6.2.6 Key Safety 173
6.2.7 Maintenance Mode 174
6.2.8 Reducing the Re-Identification Risk from Medical Data 174
6.3 Patient-Provided Medical Data 175
6.4 Privacy-Preserving Data Interface 176
6.4.1 Interface Architecture 177
6.4.2 Syntactic plugin 178
6.4.3 Differential Privacy Plugin 180
6.5 Platform Implementation and Lessons Learned 186
6.6 Related platforms 188
6.7 Conclusion 189

7 Conclusion 191
7.1 Summary of Contributions 191
7.2 Future Research 194
7.3 Final Outlook 196

A Appendix 197
A.1 Publications Related to the Thesis 197
A.2 Additional Publications 197

Bibliography 199

List of Figures

2.1 Unified taxonomy of disclosure risks 17
2.2 ICO identifiability spectrum 21
2.3 Example value generalization hierarchies 34
2.4 Taxonomy of generalization approaches 36
2.5 Domain generalization hierarchies and generalization lattices 46
2.6 Multidimensional partitioning example 47
2.7 DP intuition 58
2.8 Artificial income dataset 64
2.9 Distribution of DP counting query results 65
2.10 Distribution of DP sum query results 66
2.11 Comparison between simple and combined average mechanism 68
2.12 Adversary's initial and updated beliefs for different ε values 75

3.1 General model of SE schemes 93
3.2 Scenario of globally consistent pseudonymization 97
3.3 Setup, enroll, and revoke operations 101
3.4 Searching for and creating pseudonym operations 102
3.5 Performance of single-keyword search 108
3.6 Performance of fuzzy keyword search with varying database sizes 108
3.7 Performance of fuzzy keyword search with varying word lengths 109
3.8 Performance of fuzzy keyword write with varying word lengths 110

4.1 High-level overview of the pseudonym disclosure process 123
4.2 Overview of scheme including parties and algorithms 129
4.3 Runtimes for DKG operations 138
4.4 Runtimes for centralized key generation 138
4.5 Runtimes for encryption 139
4.6 Runtimes for encryption 139
4.7 Runtimes for combine operations 140
4.8 Runtimes for proxy key combination 141
4.9 Runtimes for scheme and message size independent operations 141

5.1 Example taxonomy trees and cut 148
5.2 First part of the SMPC circuit 154
5.3 Second part of the SMPC circuit 155
5.4 Experimental results for varying numbers of parties 158
5.5 Runtimes for different QID combinations 159
5.6 Communication demands for different QID combinations 159

6.1 Platform overview 163
6.2 Overview of cryptographic keys and encrypted data 170
6.3 Privacy-preserving data interface with abstract plugin 177
6.4 Privacy-preserving data interface for syntactic global data export 178

6.5 Privacy-preserving data interface for syntactic local data export 179
6.6 Privacy-preserving data interface for differentially private global data export180
6.7 Privacy-preserving data interface for differentially private local data export181

List of Tables

2.1 Mapping table for disclosure risks 16
2.2 Examples for pseudonymization via mapping tables 31
2.3 Example of an anatomized dataset 37
2.4 Example dataset for k-anonymity 42
2.5 2-anonymous variant of the original dataset 42
2.6 Distinct 2-diverse variant of the original dataset 44
2.7 Suspectibility of syntactic privacy models to attacks 50
2.8 Example dataset vulnerable to the homogeneity attack 51
2.9 Example dataset vulnerable to the background knowledge attack 51
2.10 Example dataset vulnerable to the similarity attack 53
2.11 Example dataset vulnerable to the minimality attack 53
2.12 Example dataset vulnerable to the downcoding attack 54
2.13 Example DP histogram query 66

3.1 Comparison to related work 114

4.1 Existing approaches and their properties 128
4.2 Overview of scheme algorithms 132

5.1 Comparison of original and updated protocol performance 157

6.1 Overview of DP frameworks 184

Acronyms

2FA two-factor authentication. 173, 189

AEAD authenticated encryption with additional data. 107, 134

AES Advanced Encryption Standard. 33, 107

API application programming interface. 175, 176, 186, 187

BDHP bilinear Diffie-Hellman problem. 94

BSI Federal Office for Information Security. 116

CCPA California Consumer Privacy Act. 26

CDHP computational Diffie-Hellman problem. 94

CHOIR Collaborative Health Outcomes Information Registry. 188

DDHP decisional Diffie-Hellman problem. 94

DKG distributed key generation. 117, 120, 123, 130–133, 136–138, 140

DP differential privacy. 4, 9, 17, 18, 26, 38, 39, 45, 56–65, 67–72, 74, 76–80, 85, 87, 88, 90, 144, 161, 162, 166, 176, 177, 180–184, 189, 192, 194, 195

DUA data use agreement. 88, 89

ECB electronic codebook. 33, 107

ECC elliptic-curve cryptography. 117, 119, 122, 130, 186

ECIES Elliptic Curve Integrated Encryption Scheme. 130

EHR electronic health record. 165

EMD earth mover's distance. 44

ENISA European Union Agency for Cybersecurity. 28

EU European Union. 12, 18, 19, 23, 25

FERPA Family Educational Rights and Privacy Act. 25, 90

FL federated learning. 78, 195

GAN generative adversarial network. 39, 78

GCM Galois/counter mode. 33

GDPR General Data Protection Regulation. 1–4, 19, 20, 23–27, 33, 90, 115, 164, 166, 168

HE homomorphic encryption. 195

HIPAA Health Insurance Portability and Accountability Act. 25, 79

HIV human immunodeficiency virus. 10, 37, 42, 44, 50, 51, 53, 54, 151

HMAC hash-based message authentication code. 32, 99, 110, 111, 173

ICO Information Commissioner's Office. 20, 21, 23, 24, 27, 212

IP internet protocol. 187

IV initialization vector. 33

KDF key derivation function. 171, 172

MAC message authentication code. 30, 32, 33, 130, 186

ML machine learning. 1, 8, 15, 18, 76–79, 82, 182, 183, 194

NIH National Institutes of Health. 188

NIST National Institute of Standards and Technology. 137, 210, 221

OSSE Open-Source-Registersystem für Seltene Erkrankungen. 189

OT oblivious transfer. 110, 111, 146

PAD peripheral artery disease. 1, 6, 37, 42, 44, 51, 53, 54, 162

PEEPLL privacy-enhanced event pseudonymisation with limited linkability. 99, 105, 110, 111, 114

PEKS public key encryption with keyword search. 93

PEP Polymorphic encryption and pseudonymisation for personalised healthcare. 188, 189

PII personally identifiable information. 19, 20, 22, 25, 84, 87, 88

PKI public key infrastructure. 136, 142

PPDM privacy-preserving data mining. 8

PPDP privacy-preserving data publishing. 8

PPRL privacy-preserving record linkage. 110–112

PRAM Post Randomization Method. 38

PRE proxy re-encryption. 5, 117, 121, 122, 127, 130, 131, 133, 135–137, 142, 193

PRF pseudorandom function. 113

PROM patient-reported outcome measure. 175

PSO predicate singling out. 90

QID quasi-identifier. 37, 40–42, 44, 46–48, 51, 54–57, 85, 144, 147–150, 156, 158–160, 167, 178–180

QR code quick-response code. 174–176, 187

RAP relaxed adaptive projection. 84

REST representational state transfer. 107, 111, 186

SE searchable encryption. 5, 91–94, 97, 99, 100, 104, 106, 114, 192

SHA Secure Hash Algorithm. 32, 175

SMPC secure multi-party computation. 6, 77, 112, 113, 115, 144–146, 152–156, 160, 193, 195

SSE searchable symmetric encryption. 92, 93

SSH secure shell. 174

TLS Transport Layer Security. 186

TOTP time-based one-time password. 173

TTP trusted third party. 28, 99, 168

US United States (of America). 18–20, 25, 87

USA United States of America. 20, 25, 26, 183

XOR exclusive OR. 146

ZKP zero-knowledge proof. 31

1 Introduction

The increased adoption of data-based statistical methods, such as machine learning (ML) techniques, enables medical researchers to gain insights that are used in evidence-based healthcare decision-making and personalized treatment strategies [BS22]. These methods can utilize a variety of personal health data including medical histories, lifestyle factors, treatment outcomes, and even genetic profiles which provide a comprehensive view of an individual's health status [RR14]. Researchers can leverage this data to, amongst others [HB20; RR14],

- identify risk factors for diseases such as cancer [CA15], diabetes [Ala+21], or peripheral artery disease (PAD) [Kre+21], and develop prevention strategies,
- examine how genetic and lifestyle factors influence individual responses to treatments, leading to patient-specific medical interventions also referred to as *precision medicine* [CV15],
- assess the impact of different treatments on various outcomes, such as recovery rates [GTC22], quality of life [Chu+22], or healthcare costs [Tri+15],
- investigate how personal factors, such as socio-economic status or geographical location, affect access to healthcare services and contribute to health inequalities [XV16], and
- improve the reality of medical care by observing and analyzing processes in day-to-day operations of hospitals [Beh+17a].

However, as the use of personal data becomes increasingly prevalent in medical research, privacy concerns and data security issues need to be taken into account progressively. The sensitivity of personal data in the domain of medical research cannot be overrated, as the data often describes some of the most private aspects of an individual's life, from genetic profiles to sensitive diagnoses [McC07; RP18]. Therefore, this data can have serious implications for an individual's privacy. This is also evident in the General Data Protection Regulation (GDPR), where Article 9 categorizes medical data as belonging to special categories of personal data, the processing of which is generally prohibited and allowed only under specific conditions.

To balance the conflict between the inherent sensitivity of personal health information and data demands by medical research, data privacy measures play a crucial role. A specific class of measures focuses on preventing the identification of individuals from data records. Pseudonymization and anonymization[1] techniques play a central role in this class of measures [NH11; VSK20]. Pseudonymization refers to the replacement of direct identifiers with pseudonyms to protect individual identities while retaining the ability to link data records and to re-identify data within a controlled environment.

[1] The terms pseudonymization and anonymization, along with the techniques they encompass, provoke intense debates. Thus, their usage in this context is for informal understanding only. A comprehensive investigation of these terms is provided in Section 2.3.

In contrast, anonymization involves the alteration of data records or data analysis processes, for example by removing identifying information, in a way that prevents linking data to specific individuals altogether. These measures can protect the identities of research participants and mitigate re-identification risks while enabling data analysis and knowledge discovery from medical data. Their importance in the field of data protection is also highlighted by the fact that they are one of the few explicitly mentioned techniques in the GDPR. By implementing robust anonymization and pseudonymization strategies into data sharing and analysis, medical researchers can balance data utility and privacy and ensure compliance with data protection regulations.

The application of these methods in itself already presents major challenges. But additionally, medical data generally is distributed over various data sources, including general practitioners, hospitals, insurance companies, and research institutions [Bey+20; Alm+19]. While this allows medical research to incorporate various aspects about a patient's health, the distributed nature poses significant challenges for the utilization of this data in medical research with respect to data integrity, security, and privacy requirements. A common strategy to utilize this distributed data in research consists in collecting the data in a central location and applying data privacy techniques before conducting data analysis. However, this strategy poses the drawback of granting the aggregating party access to unmodified, identifiable personal medical data. In scenarios where legal, ethical, or financial restrictions prevent the data from being centrally collected by a trustworthy party, this method is not a viable option [Zig+20]. Therefore, the central question that arises from these considerations is: How can we securely apply data privacy techniques in distributed environments?

In this thesis, we investigate the application of pseudonymization and anonymization techniques in distributed environments without the initial central collection of data. Other researchers recognize this area as a valuable target for future research as well [Zig+20; Car+23]. This thesis specifically focusses on the characteristics inherent to the field of medicine and utilizes illustrative scenarios drawn from this domain. However, the applicability of our contributions is not limited to the healthcare sector, but can prove relevant in comparable contexts with similar constraints, where sensitive personal data is employed for research purposes.

In the following, we present the research questions arising from this problem in Section 1.1, summarize the contributions of this thesis in Section 1.2, list publications and real-world uses of contributions in Section 1.3, and provide an overview of the thesis structure in Section 1.4.

1.1 Research Questions and Methodology

Based on the problem motivation we derive the following research questions:

RQ.1. Which risks arise from data publication of personal data and which technical measures exist to protect against these? Which advantages and disadvantages do they entail?

RQ.2. How can existing technical measures be utilized in distributed environments?

RQ.3. Can existing technical measures for distributed environments be improved with respect to specific properties?

RQ.4. How can data privacy measures be used to protect patients' privacy in distributed systems already present in medical research practice? Which requirements are dictated by their practical application?

To address the first research question **RQ.1**, we perform a comprehensive survey of existing literature and contextualize the results. The survey primarily focuses on different classes of technical measures utilized to protect individuals' data, especially their basic techniques, advantages, and disadvantages. However, the issue of privacy risks arising from data publication is equally connected to social considerations as it is to technical aspects. Additionally, the assessment of privacy-preserving techniques also is impacted by requirements derived from the application scenario as well as from regulatory frameworks. Therefore, the survey covers further related topics such as a systematization of disclosure risks, the diverse use of relevant terms in different fields, real-world examples of inadequately protected published datasets, and a subsumption of the debate surrounding data privacy across several fields.

Research questions **RQ.2** and **RQ.3** are of a general scope. This thesis aims to address them exemplarily by examining instances of various techniques employed within distributed environments. To achieve this, we have selected three prevalent examples from healthcare research, which are well-suited as representative examples.

- Pseudonymization is a data privacy measure widely used by medical practitioners and researchers due to its preservation of data utility, the option of protected re-identification, and the central role it plays in the GDPR [GS19; Mou+18]. Our first example therefore consists in the generation of pseudonyms in distributed environments where patients contribute data at multiple datasources.
- The second example considers the counterpart of pseudonym generation, that is the disclosure of patient identities behind a pseudonym. This can be an important part of pseudonymization in the medical domain, for example used to check the validity of research data [Pra21]. We investigate an approach to protect the disclosure by distributing the process across multiple parties.
- Syntactic privacy models are also used frequently in the medical domain as they preserve truthfulness on the data record level and result in datasets for which data analysts can use already existing and well-known methods and tools [DE13; Coh22]. Therefore, in our third example we examine a popular existing protocol for achieving syntactic privacy for datasets in distributed environments.

The second research question **RQ.2** represents the core of the thesis. We investigate architectures where data privacy measures are applied in distributed environments. In particular and as dictated by our examples, we look at distributed pseudonym generation, distributed pseudonym disclosure, and distributed syntactic privacy. While these architectures tackle specific problems, the encountered benefits, challenges, and limitations can provide valuable insights for the application of similar techniques in further distributed scenarios.

The third research question **RQ.3** leaves open a wide range of qualities one can look at for improving the application of data privacy measures in distributed environments. We especially focus on improving scenario-specific privacy and security properties which target the protection of individual's data or sensitive processes against various actors. Examples include the preservation of of identity confidentiality in the example of distributed pseudonym generation, an improved way to update authorized parties for distributed pseudonym disclosure, and the elimination of weaknesses in a distributed syntactic privacy protocol. Improving solutions regarding further properties, such as performance, scalability, reliability, and transparency, present valuable research opportunities as well, but are not the primary focus of this thesis.

The fourth research question **RQ.4** asks about the transfer of theoretical results into systems already used by medical research in practice. To answer this question we decided to concentrate on medical registries – systems which collect medical data regarding a specific study subject systematically over a long period of time – because they play an important role in healthcare research today [Beh+23]. We approach this question by developing a concept for a privacy-preserving medical registry using security and privacy measures including pseudonymization and anonymization techniques. This concept deals with practical problems such as authentication and authorization, secure passwords, key safety, and GDPR compliance. The implementation and application in two real-world medical studies allow us to evaluate our concept in a practical setting.

1.2 Contributions

Within this thesis, various research contributions focusing on the distributed application of privacy-preserving technical measures for data publication are presented. This section summarizes the main contributions of this thesis and explains their relation to the research questions.

C.1: Technical and Legal Literature Review

This contribution entails the results of the comprehensive literature overview. First, we look at the risks which can arise for data subjects from the dissemination of personal data and present real-world examples of published, allegedly protected datasets, whose protection measures turned out to be insufficient. After providing an overview of (legal and technical) definitions of relevant terms *personal data*, *anonymization*, *de-identification*, and *pseudonymization*, we investigate specific techniques for privacy-preserving data dissemination. We trace the development of the field coarsely and look specifically at pseudonymization, long-known de-identification techniques like generalization, syntactic privacy models, and finally semantic privacy models and differential privacy (DP) in particular as a representative for the state-of-the-art. This contribution does not focus on the distributed setting but just introduces fundamental techniques in detail. The debate around data privacy and (un-)suitable technical measures caused emotional discussions, especially in the fields of computer science and law. Therefore we finally give an overview of different perspectives in this debate, try to clear up the underlying misunderstandings, and provide an outlook on potential solutions to the debate. The contribution may be of independent interest to readers who want to gain insight into the field of privacy-preserving data dissemination as a foundation for future research beyond the scope of this thesis. This contribution is directed at research question **RQ.1**.

C.2: Distributed Pseudonym Generation

We present a solution for the generation of pseudonyms in distributed environments. Data records regarding indiviuals can be collected at various data sources. A data processor relies upon these data records but should only ever receive pseudonymized data records to limit privacy risks. The main objective is to support *globally consistent pseudonyms*, in other words each individual is consistently assigned the same pseudonym regardless of the originating data sources. Building up on related work, we propose several properties which a solution should entail at best. With these properties in mind, we present our own scheme which utilizes a specific searchable encryption (SE) scheme to especially tackle the problem of managing data sources, which was left open in related work. We compare our scheme to related work with respect to proposed properties and find that no scheme fulfills all properties, leaving room for further research work. This contribution addresses research question **RQ.2** in that we provide an approach to distributed pseudonymization. Additionally, it also tackles research question **RQ.3** in that we improve existing solutions to the problem of distributed pseudonym generation regarding the management of data sources.

C.3: Distributed Pseudonym Disclosure

Additionally to the distributed generation of pseudonyms we present an approach to distribute the pseudonym disclosure process. Due to the sensitivity of the data subject-pseudonym relationship this process should be protected at all costs. One method to protect sensitive processes is the so-called *multi-eye principle* – the idea to require the approval and action of multiple parties to perform a process. Our approach implements the multi-eye principle for pseudonym disclosure based on a cryptographic scheme referred to as *threshold decryption*. These schemes distribute the decryption process in public-key schemes so that multiple authorized parties have to collaborate for the decryption of ciphertexts. While applying these schemes to protect pseudonym disclosure is straightforward, further challenges arise in practice. Especially the question of key management, for instance how to enable or disable parties to participate in the decryption process, is not trivial. We propose a novel approach to this problem based on combining threshold decryption and another cryptographic scheme called proxy re-encryption (PRE) to improve aspects of the key management in comparison to related work. This contribution addresses research question **RQ.2** in that we provide an approach to distributed pseudonym disclosure. Furthermore we improve on existing approaches for key management in threshold schemes (**RQ.3**).

C.4: Distributed Syntactic Anonymization

In this contribution we shift the focus from distributed pseudonymization to distributed anonymization. In particular, we analyze a specific protocol for distributed syntactic privacy, uncovering weaknesses that compromise the central syntactic privacy guarantees of the protocol. These weaknesses arise from the unthoughtful use of a subprotocol for secure summation of sensitive inputs. We employ a general secure multi-party computation (SMPC) framework to replace the subprotocol and prevent the weaknesses. Using the framework requires a deep analysis of the original protocol, a thoughtful construction of computation circuits essential for the SMPC subprotocol, and an adequate substitution of the vulnerable subprotocol. This contribution addresses research questions **RQ.2** and **RQ.3** in that we provide an updated approach to distributed syntactic anonymization which prevents vulnerabilities in the original approach.

C.5: Privacy-Preserving Medical Registry

In comparison to primarily academic contributions discussed so far, this contribution focuses on the practical application of privacy-preserving techniques to improve privacy and security guarantees in systems prevalent in medical research. In particular, we target the problem of designing and implementing a privacy-preserving medical registry – a system to prospectively collect data concerning patients which match certain criteria over a longer period of time. The registry should offer further functionalities such as monitoring the validity of collected data and comparing the performance of healthcare providers. We implement a distributed system which implements various privacy and security mechanisms to protect personal and medical patient data. Additionally, we offer an interface enabling researchers to securely access medical data with adequate privacy protections in place. This contribution tackles research questions **RQ.2** and **RQ.4** by concentrating on the practical implementation and deployment of a privacy-preserving technical solution in a distributed environment and comprehensively detailing the essential measures required to meet security and privacy requirements.

1.3 Publications and Real-World Use

The central motivation of this thesis and preliminary ideas for contributions **C.2**, **C.3**, and **C.5** have been published in the proceedings of the *GI Sicherheit* [Pet20]. Contribution **C.3** extends work published at the *ACM SAC* [Zim+20]. The general platform concept for the privacy-preserving medical registry (contribution **C.4**) has been published in *DuD* [Pet+19]. The developed platform is used in two medical studies. The *IDOMENEO* study [Beh+17a] examined the reality of care for patients suffering from PAD in over 30 medical centers in Germany. As part of the study, data from over 5,600 patients was collected in our platform [BD21]. The ongoing *INCREASE* study [Klo+22] investigates the use of modern therapy concepts in minimally invasive heart valve procedures.

1.4 Structure

The thesis is structured as follows: Chapter 2 presents the results of the literature review and therefore covers contribution **C.1**. In Chapter 3 we introduce our approach to the generation of globally consistent pseudonyms in distributed environments (contribution **C.2**). Our method for achieving the multi-eye principle for pseudonym disclosure (contribution **C.3**) is covered in Chapter 4. Chapter 5 introduces our contribution **C.4**: it demonstrates the vulnerabilities in a protocol for distributed syntactic anonymization and our updated protocol which prevents these vulnerabilities. In Chapter 6 we present a technique for privacy-preserving medical registries (contribution **C.5**). Chapter 7 concludes the thesis.

2 Data Privacy Measures for Protecting Sensitive Data

The importance of privacy-preserving publishing and processing of sensitive personal data has been acknowledged for a long time. A substantial body of research literature, practical guidelines, and legal regulations have emerged from different disciplines, including statistics, law, computer science, and even philosophy. In this chapter, we present a comprehensive overview of the achieved results (and failures). While our primary focus is on the various technical measures employed to protect individuals' data in datasets, we include further related topics as well. These include the risks associated with data publication, an overview of relevant terms and their different understandings, real-world examples of inadequately protected published datasets, and a subsumption of the debate surrounding data privacy across several fields.

Our overview concentrates on so-called microdata, that is information about specific individuals on an individual level [BS08]. One can think of this as data in a tabular form, in which attributes are represented as columns and each individual is associated with one (or multiple) rows. We explicitly do not address privacy risks arising from "rich" data types, such as images, videos, audio, or unstructured text. Readers interested in these topics can look into several surveys [RAP16; CMV21; ZC22] as a starting point for further research.

In the following, we briefly introduce fundamental concepts related to data privacy as a basis for the chapter. There are two main research directions in the area of privacy-preserving usage of data: privacy-preserving data publishing (PPDP) and privacy-preserving data mining (PPDM). PPDP deals with the processing of datasets in a way so that they can be published as individual records without violating the privacy of included individuals. There are several surveys regarding PPDP [Che+09; Fun+10a; GLS14; Zig+20; Ola+22]. PPDM, a term coined by Agrawal and Srikant [AS00], comprises methods that extract knowledge from datasets while preventing the disclosure of individuals' sensitive information. These methods can vary from publishing statistical tables with aggregated values over classical data mining methods like association rule mining to the privacy-preserving application of modern ML techniques such as neural networks. There are several surveys about results in this field [AY08; Agg15; MV17].

The distinction between these two fields is not always clear. While some publications see PPDP as a subcategory of PPDM [MV17; Alp16; AY08; Agg15], others clearly differentiate between them: Zigomitros et al. [Zig+20] highlight the importance of truthfulness on the record level for PPDP methods which is not the case for PPDM. Another difference they bring up is that PPDP methods in most cases do not target specific types of data analyses but allow for the application of various techniques with different objectives. This requires to protect the privacy of individuals before releasing the data against all possible attacks. In comparison, in PPDM scenarios the query must be known before applying privacy-preserving measures and necessary safeguards specific to the data mining task can be applied during method execution.

A related distinction is the one between *interactive* and *non-interactive* privacy mechanisms [Dwo06]. In the interactive setting, the trusted dataset holder provides a query interface for authorized users. To protect individual's privacy, one can use query auditing, that is, deny queries which reveal sensitive information, or output perturbation, in other words, perturb query results in a privacy-preserving way [Agg+05]. In the non-interactive setting, the dataset is transformed in a way that aims at protecting the privacy of individuals. An important subcategory is what Ohm [Ohm09] terms *release-and-forget* anonymization[1]: The dataset gets anonymized, for example, by removing direct identifiers and perturbing others, and is afterwards released publicly without further restrictions. Other possibilities include publishing aggregate statistics or subsamples of the full dataset.

Finally, we distinguish two classes of privacy models based on their underlying principle. A class of related models can be subsumed under the term *syntactic privacy models*[2]. They guarantee privacy by syntactic conditions on the structure of the dataset, for example, by generalizing records in a dataset until an algorithmically verifiable condition is met [CT13]. One example (we will cover in Section 2.6.2) is k-anonymity – a privacy model which requires that at least k data records in a dataset share the same set of a specific kind of attribute. This syntactic condition can easily be checked by just looking on the dataset.

In comparison, *semantic privacy models*[3] (sometimes also referred to as *probabilistic privacy models*) are concerned with the relationship between inputs and outputs of data releasing mechanisms [KM12]. These models describe a property of the mechanism and not a property of its result. The most famous representative of this model class is DP, which we will introduce in Section 2.7. The basic idea of differentially private mechanisms is to release results on which the presence (or absence) of a single individual in the dataset has only small and provably bounded influence.

After having introduced these preliminary concepts, we provide an overview of this chapters's content and structure. The first Section 2.1 deals with privacy risks arising for individuals from datasets. Section 2.2 presents a fundamental tradeoff between privacy and utility when using privacy techniques for datasets. In Section 2.3, we look into definitions of relevant terms like *personal data*, *pseudonymization*, and *anonymization*. The following sections cover relevant privacy techniques, including their functionality, strengths, and weaknesses. In this chapter, we only cover generally applicable methods, that is methods which can be used for a variety of data analysis tasks, but not specific privacy-preserving methods for a distinct task (such as the privacy-preserving computation of association rules [Ver13]). We present details on Pseudonymization in Section 2.4, on simple de-identification techniques in Section 2.5, on syntactic privacy models in Section 2.6, and on DP as a representative semantic privacy model in Section 2.7. Section 2.8 provides an overview of practical re-identification attacks. The

[1] The term *anonymization* and related terms like de-identification, while inducing some intuitive idea of their meaning in most people, entail quite varying understandings. We provide an overview in Section 2.3.

[2] The differentiation between syntactic and semantic privacy models probably was first introduced by Machanavajjhala [Mac08].

[3] Here, the term *semantic* is borrowed from *semantic encryption*, introduced by Goldwasser and Micali [GM84]: "Informally, a system is semantically secure if whatever an eavesdropper can compute about the cleartext given the ciphertext, he can also compute without the cyphertext."

final Section 2.9 portrays the academic discourse in computer science and law about data privacy and appropriate techniques, integrates the contents of this chapter in the discourse, and looks into proposals for necessary policy changes.

2.1 Systematization of Disclosure Risks

Publishing or sharing data can entail different risks for individuals, generally referred to as *disclosure risks* (other publications refer to these risks as privacy threats [GLS14]). Solove [Sol06] defines disclosure as "the revelation of truthful information about a person that impacts the way others judge her character". This information can cause physical, emotional, financial, or reputational harm in varying ways, including [Sol06]:

- The information can put people at direct risk, for example, in the case of published addresses of domestic abuse victims.
- The information can lead to irrational judgment of an individual based on common stereotypical opinions with respect to, amongst others, diseases such as human immunodeficiency virus (HIV), political views, sexual orientation, and socioeconomic status.
- It can cause discriminatory and societally undesirable decisions, for example, employment decisions based on genetic data.
- Scattered information about an individual can distort the assessment of the individual by others.
- Published information about an individual's past can inhibit their ability to change for the better if the stigma of their past actions sticks to them.
- Public information can be used in a variety of unexpected ways, including many which deviate from the implicit or specified purpose the data should fulfill.

These risks can harm individuals contributing to published or shared data especially, but in some cases even other individuals.

In this section, we have a closer look at the disclosure risks individuals may face from having their information published. Existing literature considers differing sets of risks, names risks in different fashions, or uses the same term for different risks or as Duncan and Lambert express it, "different interpretations of disclosure from microdata are possible and confusion is likely as long as intuition is not formalized." [DL89] In this section we provide an overview of these varying risk definitions and develop a unifying taxonomy of disclosure risks.

2.1.1 Literature Overview

A first disclosure concept was provided by Dalenius [Dal77]: "If the release of the statistics S makes it possible to determine the value D_K more accurately than is possible without access to S, a disclosure has taken place". D_K in this case describes an arbitrary attribute value (even for attributes not in the dataset) for some individual (not necessarily part of the dataset). In other words, according to Dalenius, a disclosure takes place as soon as the dataset alters an adversary's beliefs about any individual's data to any extent. This is the broadest definition of a disclosure imaginable. Furthermore, Dalenius introduces ideas about the (un-)certainty of a disclosure by differentiating between *exact* and *approximate* disclosure.

To the best of our knowledge, the first conceptualization of different disclosure risks was provided by Duncan and Lambert [DL89]. They distinguish four different risks: identity disclosure, attribute disclosure, inferential disclosure, and population disclosure. *Identity disclosure* and *attribute disclosure* term related risks: Identity disclosure describes the association of an individual with a data record and attribute disclosure the possibility to obtain reliable information about an individual as a result of this association. They use *inferential disclosure* as the risk for inferring more information about an individual with high confidence than what could be determined without the dataset. Finally, they include the risk of *population disclosure* (also referred to as *model disclosure*). This describes the possibility to draw conclusions about the (confidential) relationship between population characteristics and sensitive attributes.

Machanavajjhala et al. [Mac+06] cover *attribute disclosure* in detail. They include the positive and negative disclosure principles. A *positive* disclosure occurs if an adversary can deduce the value of a sensitive attribute for some individual. In contrast, a *negative* disclosure allows an adversary to rule out one or multiple sensitive attribute values for an individual. Additionally, they explicitly use a probabilistic formulation, namely "an adversary can correctly identify the value of a sensitive attribute with high probability" [Mac+06].

Nergiz, Atzori, and Clifton [NAC07] introduce the concept of *membership disclosure* which tells an adversary if an individual's data is (not) part of the published dataset. This information can already present a severe privacy violation, for example, when the presence of an individual in a database of cancer patients can be inferred.

Fung et al. [Fun+10a] categorize disclosure risks into *record linkage*, *attribute linkage*, *table linkage*, and *probabilistic attack*. *Record linkage* describes the unique identification of an individual's data record in the dataset and *attribute linkage* the inference of sensitive values of an individual from the dataset (without precisely identifying the data record). *Table linkage* represents the risk which allows an adversary to determine the presence or absence of an individual's data record in the dataset. They extend these risks with the *probabilistic attack*: This risk exists, when the prior and posterior beliefs of an adversary about an individual's sensitive information in the dataset vary to a large extent caused by the published dataset.

Another slightly different variant of definitions is given by Templ [Tem17]. They use *identity disclosure* as usual, in other words, as the association of an individual with a specific data record. *Attribute disclosure* is described as the risk which allows an intruder to determine new characteristics of an individual based on the information in the dataset. They categorize *membership disclosure* as a special case of attribute disclosure which describes the disclosure of group membership, for example, when the individual is part of a community of faith. Furthermore, they introduce *inferential disclosure* as the possibility to determine some sensitive attribute value of an individual more accurately with the dataset information than without.

The European Union (EU) Article 29 Working Party describes three risks the anonymization of data should protect against in their *Opinion 05/2014 on data anonymization techniques*: singling out, linkability, and inference [Par14]. *Singling out* means to find records identifying an individual in the dataset. *Linkability* allows an adversary to link records of an individual or at least a group of individuals in the same or different datasets – a risk, which is not explicitly covered by the previous risk definitions. Finally, *inference* is the possibility to deduce an attribute value from a set of other attribute values.

The ISO standard 25237 *Health informatics — Pseudonymization* [Sta17] also provides an overview of disclosure risks (referred to as attacker goals). The standard differentiates between full re-identification, partial re-identification (information recovery), and database membership. *Full re-identification* describes the association of individual and data record and further divides this goal regarding the direction of identification: An adversary's goal can be to identify the individual related to a specific data record or to identify the data record for a specific individual. *Partial re-identification* means the inference of single characteristics for individuals from the dataset. *Database membership* stands for the determination of (non-)participation of an individual in the dataset. It explicitly mentions the statistical nature of algorithms used to reach these goals.

The ISO/IEC standard 20889 *Privacy enhancing data de-identification terminology and classification of techniques* [Sta18] also differentiates re-identification attacks based on their goals:

- Re-identify a record belonging to a specific individual (*prosecutor attack*),
- re-identify the individual of a specific record (*journalist attack*),
- re-identify as many records with the corresponding individuals (*marketer attack*),
- establish the presence of an individual in the dataset, and
- deduce a sensitive attribute associated with a group of other attributes.

The standard includes the risks from [Par14] as *re-identification approaches* used to reach the attacker goals.

2.1.2 Unified Taxonomy

In this section, we provide a unifying taxonomy and categorize the covered disclosure risks discussed in the literature within it.

First, we specify the scenario to allow for unambiguous disclosure risk definitions. A dataset D contains n data records $D_1, \dots, D_n$ and each data record consists of the same m attributes $A_1 \in \mathcal{A}_1, \dots, A_m \in \mathcal{A}_m$ for attribute value domains $\mathcal{A}_i$. Each data record belongs to exactly one individual $I \in \mathcal{P}$ for some population $\mathcal{P}$ (while there can be several data records belonging to the same individual, potentially). In most cases, there are individuals $I \in \mathcal{P}$ who are not part of the dataset D, so $\mathcal{P} = \{I \in D\} \cup \{I \notin D\}$[4]. As an example, one can think of the *German Cancer Registry* providing data like gender, age,

[4] We abuse our notation slightly here. $I \in D$ should be understood as *dataset D contains at least one record D_i which is related to individual I.*

city, cancer diagnosis, and several medical information about individual cancer patients. All these patients are part of the larger population of German citizens.

Identity, Attribute, and Membership Disclosure

With this scenario in mind, we can develop our unifying taxonomy. In the literature [DL89; Fun+10a; Sta18; Sta17] there are three main disclosure risks related to individuals $I \in D$:

- An adversary might be able to link a data record $D_i \in D$ to an individual I_j, disclosing all attribute values to them. We follow Duncan and Lambert [DL89] here and use the term *identity disclosure*. Some authors [Sta18; Sta17] further divide this risk depending on the goal of an attacker, but the underlying risk of linking data record and individual remains the same.
- An adversary might be able to deduce some attribute values of an individual I_j from the dataset D (without directly identifying the specific data record D_i). For this risk we use the term *attribute disclosure*, like, amongst others, Templ [Tem17] does.
- An adversary might be able to tell, if an individual $I \in \mathcal{P}$ is part of the dataset D. As in the publication introducing this risk [NAC07], we term this disclosure risk *membership disclosure*.

Deterministic and Probabilistic Disclosure

Each of these disclosure risks can be either considered in a deterministic or in a probabilistic sense. For the deterministic case, a disclosure happens only if the adversary is certain about the specific deduction, for example, if they can link the data record to an individual with probability $\mathbb{P} = 1$. This interpretation is especially common for the case of identity disclosure and membership disclosure.

A probabilistic disclosure, on the other hand, does not require this certainty. There are two possible interpretations of this disclosure type:

- First, the disclosure allows an adversary to deduce the disclosing fact with high certainty. One example for this interpretation is given by Machanavajjhala et al. [Mac+06] who use this interpretation for attribute disclosure.
- The second interpretation defines the disclosure as the significant change of an adversary's beliefs about the disclosing fact based on the dataset. This is described by Fung et al. [Fun+10a] as a *probabilistic attack* and is also related to Dalenius' disclosure definition [Dal77].

While we do not distinguish these cases in our taxonomy to keep it more general, this distinction can lead to strongly differing disclosure scenarios. For example, if a dataset D changes an adversary's belief about an individual I suffering from a specific disease c from a prior belief of $\mathbb{P}[D^I_{disease} = c] = 0.01$ to a posterior belief $\mathbb{P}[D^I_{disease} = c \mid D] = 0.2$, this might not be considered a high certainty in most cases, but surely it changes the adversary's beliefs to a large extent. Furthermore, it should be noted that the terms *with* *high certainty* and *to a large extent* allow for a wide spectrum of concrete instantiations of what is defined as a disclosure. Finally, we want to mention that while identity disclosure in the literature most often refers to deterministic identity disclosure, attribute disclosure is often regarded in a probabilistic sense.

Positive and Negative Disclosure

Another distinction, often neglected in the literature, is the one between positive and negative disclosure, as introduced by Machanavajjhala et al. [Mac+06] for attribute disclosure. A disclosure can not only describe the situation in which an adversary can determine the specific attribute value A_i for an individual I (*positive attribute disclosure*), but also the situation in which they can eliminate possible attribute values (*negative attribute disclosure*). For example, when an individual is not attributed as *free from chronic diseases* in an insurance dataset, this sole information poses a risk for the individual – independent from the specific chronic disease the individual suffers from.

Negative Membership Disclosure

While the risks covered so far are related to individuals $I \in D$ whose data is part of the dataset, in specific situations a dataset can also pose disclosure risks for individuals $I \notin D$ not in the dataset. Analogous to the case of negative attribute disclosure, we include negative membership disclosure in our taxonomy. The fact that an individual's data is *not* part of a dataset can also pose a risk for affected individuals. For example, not being part of a donors dataset can lead to social pressure for the individual.

Population Disclosure

Another rarely covered disclosure risk results from deductions an adversary can draw from statistical or deterministic correlations in the dataset D which are valid in the whole population $\mathcal{P}$. These correlations learned from dataset D can also be used to harm individuals $I \in \mathcal{P}$ whose data is not contained in dataset D, so that publishing the dataset D results in a disclosure risk for individuals $I \notin D$. For example, when an adversary learns from a dataset that specific markers in the human genome are responsible for an individual's probability of suffering a depression, they can exploit this fact against all individuals from the population, not just ones participating in the dataset. Another example includes the refusal of loans or varying insurance costs for individuals from a certain subpopulation based on statistical inferences with respect to the subpopulation drawn from a dataset in which the respective individual's data is not contained. Following Duncan and Lambert [DL89], we call this risk *population disclosure*. Since this disclosure risk for individuals I in the dataset D is already covered by our notion of probabilistic positive attribute disclosure, we explicitly *exclude* individuals $I \in D$ in this disclosure risk.

One can argue that drawing conclusions about individuals whose data is not part of the dataset D can barely be seen as a disclosure risk related to D. However, Cormode [Cor11] takes a different line of thought here. He argues that it is irrelevant to an adversary whether an individual is part of a dataset if it allows them to learn individual's sensitive information with a high probability – or as they put it: "The core issue is that latent properties of a population, when learned, can compromise the privacy of an individual" [Cor11]. But, as they additionally state, this issue might not be resolvable (at least in some settings) since learning statistical correlations can represent the exclusive reason for the existence of datasets.

Related to this disclosure risk is the discussion about the deFinetti attack covered in Section 2.6.8, where a ML classifier is trained on a syntactically anonymized dataset to disclose correlations between attribute values of an individual and their sensitive values. With respect to this attack, Clifton and Tassa [CT13] argue that it should only be considered a threat to an individual's privacy if the success of the attack, in other words, the accuracy of the trained model, is significantly higher for individuals whose data is part of the database (which we define as probabilistic attribute disclosure).

El Emam and Álvarez [EÁ14] provide similar arguments regarding population disclosure in general. While "[i]nferences from data can be discriminatory, stigmatizing, creepy, or surprising" [EÁ14] and decisions based on these inferences can have serious consequences for individuals, this is independent of their participation in the dataset. In the authors' view, the problematic part are not the inferences themself but the (potentially inappropriate) decisions based on them. Similarly, Schwartz and Solove [SS11] rate this risk from a legal standpoint as something which is broader than just privacy law. Regulations concerning, for example, civil rights, discrimination, and insurance policies play a decisive role. This argument is based on the idea that privacy harms result from the use of data associated with specific individuals.

Resulting Taxonomy

Combining these ideas results in a taxonomy of disclosure risks presented in Figure 2.1. Table 2.1 provides a mapping of the terms used in literature and our taxonomy. After looking at the fundamental disclosure risks arising from data publication, in the next section we present the fundamental problem of privacy-preserving data publication.

2.2 Privacy-Utility Tradeoff

A central concept in the field of privacy-preserving data mining and publishing is the *privacy-utility tradeoff*[5]. The concept describes the conflict of protecting the privacy of individuals in a dataset and the feasibility to compute useful statistics based on this dataset. Protection methods can balance these two opposing goals, but achieving both is generally not possible. The existence of such a tradeoff for these methods is intuitively obvious by looking at two extreme cases [RSH07]. Publishing an unaltered version of a dataset provides maximum utility and no privacy. In contrast, not publishing the dataset at all provides maximum privacy without any utility.

[5] In the literature nearly each and every combination of *privacy, risk, or protection* and *utility, accuracy, or benefit* can be found as term for this tradeoff concept.

Table 2.1: Mapping table between the terms for disclosure risks used in literature and our proposed unifying taxonomy. The subheadings describe *deterministic* (D) or *probabilistic* (P) risks, respectively.

		Identity Disclosure		**Attribute Disclosure**				**Membership Disclosure**				**Population Disclosure**	
				Positive		Negative		Positive		Negative			
		D	P	D	P	D	P	D	P	D	P	D	P
[DL89]	Identity Disclosure	x											
	Attribute Disclosure	x											
	Inferential Disclosure				x								
	Population Disclosure												x
[Mac+06]	Positive Disclosure				x								
	Negative Disclosure						x						
[NAC07]	Membership Disclosure								x				
[Fun+10a]	Record Linkage	x											
	Attribute Linkage				x								
	Table Linkage								x		x		
	Probabilistic Attack				x								
[Tem17]	Identity Disclosure	x											
	Attribute Disclosure			x									
	Membership Disclosure			x									
	Inferential Disclosure				x								
[Sta17]	Full Re-identification	x											
	Partial Re-Identification			x									
	Database Membership							x					
[Sta18]	Prosecutor Attack	x											
	Journalist Attack	x											
	Marketer Attack	x											
	Presence Establishment							x					
	Sensitive Attribute Deduction				x								

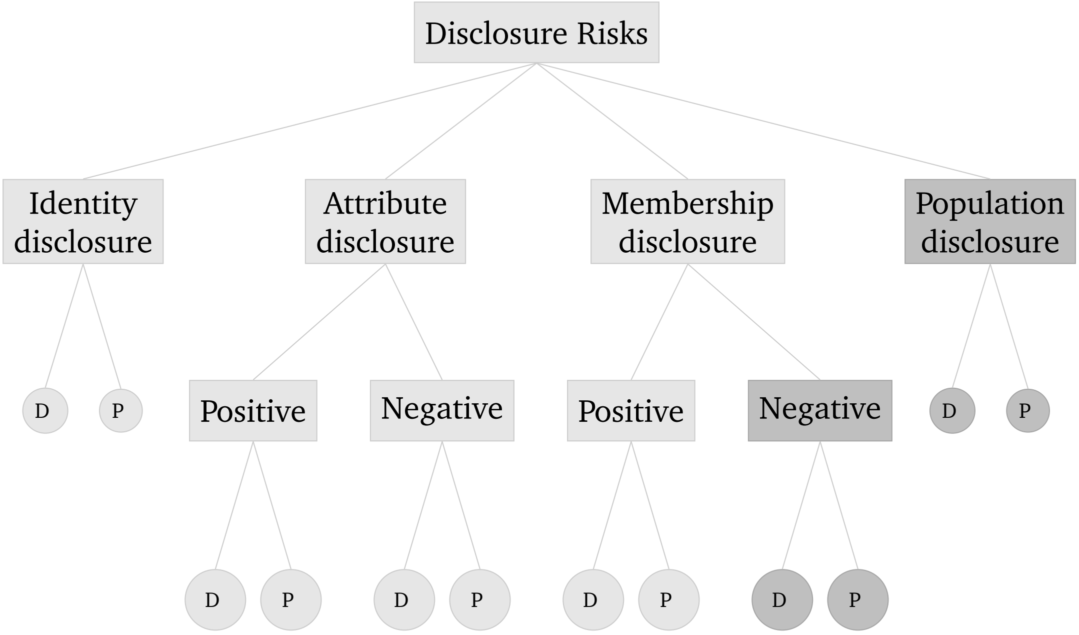


Figure 2.1: Unified taxonomy of disclosure risks. The leaf nodes describe *deterministic* (D) or *probabilistic* (P) risks, respectively. While identity, attribute, and positive membership disclosure apply to individuals whose data is part of the dataset, inferential and negative membership disclosure apply to non-participants (indicated by the different shade of gray).

Li and Li [LL09] point out the differences between privacy and utility and their indirect relationship: On one hand, publishing data can provide utility to the whole society, for example, by allowing medical research to use precise data, and each provided data record potentially increases the utility. On the other hand, the publication can result in privacy loss for individuals and affect them to different degrees. So privacy is an individual concept and the publication of a dataset must depend on protecting each individual, while utility is an aggregate concept and published information adds up, even when distortion reduces the utility.

When looking at concrete instantiations of the privacy-utility tradeoff, one has to choose specific measures for both concepts. For privacy, options include the privacy guarantees of syntactic privacy models like k-anonymity (see Section 2.6) or comparisons of prior and posterior adversarial beliefs like used in DP (see Section 2.7). There are two generic approaches for measuring the utility of a dataset. First, one can use generic metrics to express the general utility of a protected dataset in numerical terms. Since there is no direct metric for general utility, often a metric based on information loss with respect to some aspect of the protection process is used as a proxy [BS08]. The basic idea is that less distorted datasets generally provide better utility. Examples include the number of generalization steps necessary to obtain the protected dataset or averages sizes of resulting equivalence classes [LL09]. Brickell and Shmatikov [BS08] argue that the utility of a dataset must always be assessed with respect to a specific data mining task. Protected datasets might support the accurate execution of one data mining task, for example, classification according to a sensitive attribute, while not supporting others, such as clustering based on another attribute. This leads to the second approach which assesses the utility of a dataset in terms of a specific data mining task. For example, Rastogi, Suciu, and Hong [RSH07] use the estimation of counting query results as a utility measure. Li and Li [LL09] counter this approach by arguing that if the specific task is known beforehand, one can just publish the result of this task instead of the protected dataset.

The general approach to this tradeoff in practice is to (intuitively) choose a privacy requirement, for example, a specific syntactic privacy model with required parameters, and to generate a protected dataset afterwards, which meets the requirement and maximizes a predefined utility measure [LL09].

However, several publications try to formalize the complete space of solutions to the privacy-utility tradeoff with differing approaches. Li and Li [LL09] employ an economic concept from the *Modern Portfolio Theory* which can be used to guide financial investments and base their approach on the similarity of the privacy-utility tradeoff and the expected-return-risk tradeoff. Brickell and Shmatikov [BS08] balance the tradeoff between privacy in terms of adversarial sensitive attribute disclosure and utility in terms of concrete ML tasks. Loukides and Shao [LS08] focus on k-anonymity providing an optimal privacy-utility tradeoff. Sankar, Rajagopalan, and Poor [SRP13] provide an analytical model based on information theory to guarantee tight bounds on the achievable optimal privacy-utility tradeoffs. Several publications deal with the privacy-utility tradeoff in DP, amongst others [Alv+12; KL10; KM11; Nan+22]. In this model, the privacy parameter ε can be considered as an (unintuitive) way to balance privacy and utility. Recently, it has been shown that the same tradeoff exists in synthetic data generation (see Section 2.5.9).

After covering this fundamental tradeoff in privacy-preserving data publication, we turn our attention to a discussion of relevant terms in this field, such as *pseudonymization* and *anonymization*.

2.3 Term Definitions

There often are quite different understandings of relevant terms in the field of the privacy-preserving publication of data. Depending on the background of the person, terms like *de-identification*, *anonymization*, and *pseudonymization* can describe many different concepts and techniques. Additionally there are widespread misunderstandings of these terms in the general public. One common example is the equation of pseudonymized and anonymized data. In fact, this misconception occurs so often, that a lot of publications dealing with anonymization or pseudonymization contain sections about the difference between pseudonymization and anonymization [ENI18]. To clear up these misconceptions, in this section we want to provide an overview of the understanding of the relevant terms *de-identification*, *pseudonymization*, and *anonymization*. Covering the most relevant fields, we present views popular in the technical as well as legal community[6].

[6] We neither have the time resources nor the means to investigate all privacy regulations and discussions worldwide. Therefore we concentrate on the perspective of the EU, which is not only our origin but also has one of the most advanced data privacy regulation, and the United States (of America) (US), which is the home of not only some of the most influential technology corporations but also of several legal scholars who have provided central contributions to the privacy debate.

2.3.1 Personal Data and Personally Identifiable Information

As a basis, we first have to introduce the concept of *personal data*, used in the GDPR, or personally identifiable information (PII), the central concept in most US privacy laws. Informally, these terms describe whether some data is related (or relatable) to a specific individual. For a large range of data protection regulations these principles serve as a defining factor for the scope and boundaries of the respective regulation [SS11]. If data meets a law's definition of personal data or PII, the law applies and restricts the data collection, processing, or disclosure. Otherwise, the law offers no protection for the data.

This concept is especially relevant for anonymization, since anonymizing the data frees it from regulatory burdens and allows, for example, the unregulated dissemination of anonymized datasets.

The EU Perspective: Personal Data

Personal data is defined in the GDPR, Article 4:

> 'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;

Here, not only data with respect to *identified* persons is covered, but also data concerning potentially identifiable persons. Recital 26 of the GDPR provides more details about the aspect of identifiability. For this, one has to consider "all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly". More details for the term *reasonably likely* are given. For assessing the likeliness of identification, according to the GDPR one has to incorporate "all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments."

The EU *Article 29 Data Protection Working Party* in their *Opinion 05/2014 on Anonymisation Techniques* [Par14][7] provides three key indicators for determining if some information should be considered identifiable regarding an individual, in other words personal data:

- *Singling out*: Is it possible to isolate some or all data records concerning an individual in a dataset?
- *Linkability*: Is it possible to link data records concerning the same individual (or a group of individuals) in a database or across databases?

[7] The Working Party refers to EU *Directive 95/46/EC*, the predecessor of the GDPR. But most of the content of Recital 26 is directly transferred to the GDPR. Their results are generally referred to in the GDPR context as well.

- *Inferences*: Is it possible to predict some attribute values of an individual from the dataset with significant probability?

The Information Commissioner's Office (ICO) argues that identifiability may be viewed as a spectrum and the classification of the identification risk depends on the circumstances of the data processing [ico22]. They provide more details on factors to consider when assessing the likeliness of identification. Special attention should be paid to the release model, such as public release or release to specific groups. For each release, one should consider the data itself, the context, scope, and purpose of its processing, as well as applied technical and organizational measures. Additionally, one should incorporate the motivation for identification, the required competence and further objective factors. They further emphasize that the understanding of identifiability is based on the likeliness criterion. It is not required to consider every purely hypothetical possibility for identifying an individual, in other words, to prevent the risk of identification under all circumstances, but just what is *reasonably likely*. Figure 2.2 provides a visualization of the risk spectrum as envisioned by the ICO.

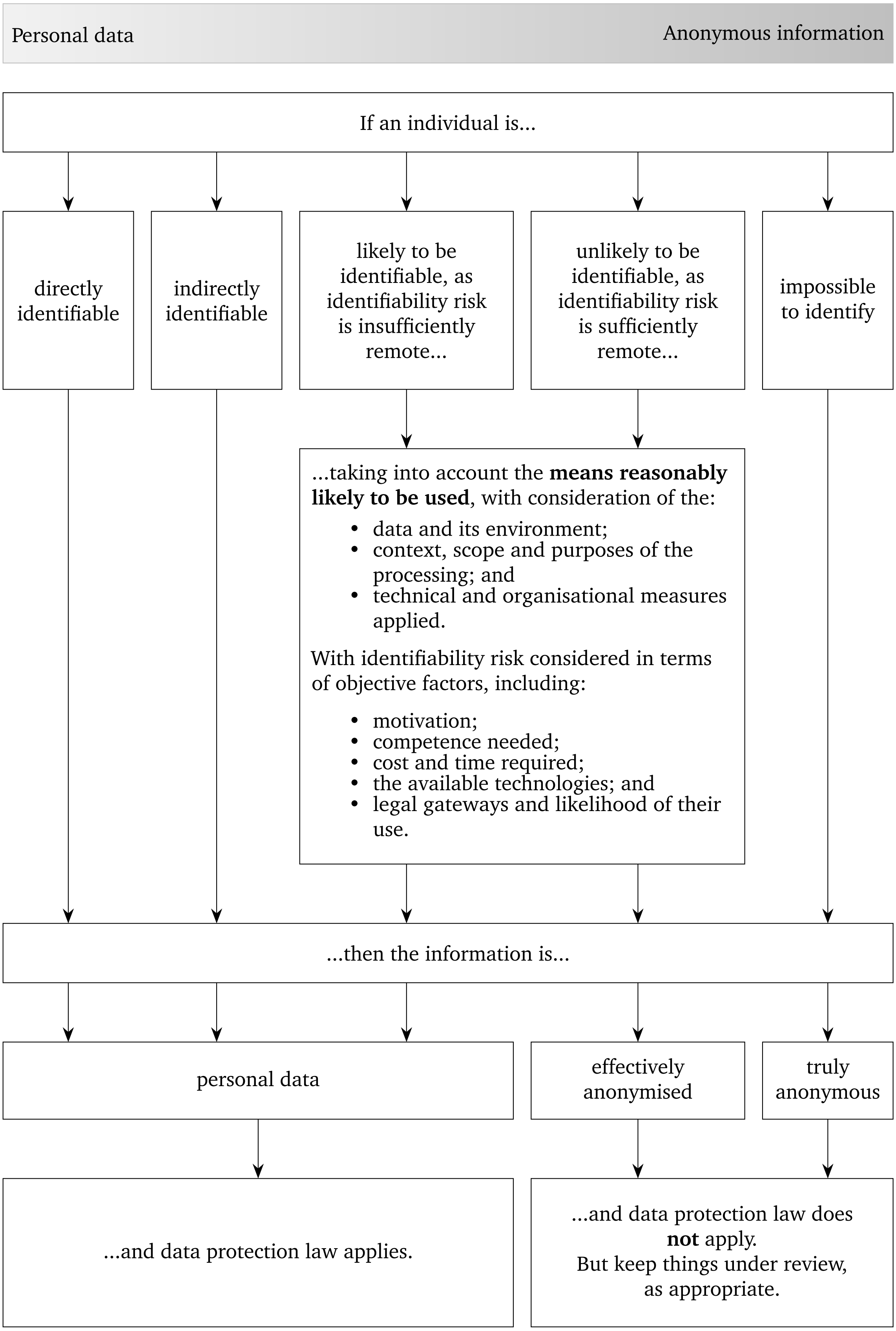
The US Perspective: PII

In the United States of America (USA), the related concept of PII is used. But in comparison to the term personal data, which at least has a distinct definition in the GDPR, there is a variety of PII definitions in privacy law and no consensus between scholars [Nis+17]. Schwartz and Solove [SS11] differentiate between three predominant approaches for definition:

- The *tautological* approach defines PII as any information that identifies a person or, as Schwartz and Solove express it: "At its core, this approach simply states that PII is PII" [SS11]. This is an unhelpful definition for the practical purpose of determining if data should be considered PII.
- The *non-public* approach defines PII as being data which is not not publicly accessible and not purely statistical. But this definition is not helpful as well as it does not incorporate the principle of identifiability at all.
- The *specific-types* approach consists in listing specific types of data that consitute PII. However, this approach provides no method to determine whether a type of information should be considered PII, and therefore be part of the list, or not.

Schwartz and Solove argue, that determining if some data should be considered PII is complex and "abstract determinations of whether a given piece of information isPII are insufficient" [SS11] due to the context-dependability of identifiability. Additionally, the frequent (re-)identification of individuals in datasets assumed to be non-PII (compare Section 2.8) impede this determination even further. Nissim et al. [Nis+17] provide similar arguments. The case-by-case determination of whether data identifies an individual, and thus constitutes PII, is complicated by advanced analytical capabilities, more available personal data and sophisticated scientific understanding of privacy risks.

Ohm [Ohm09] argues for abandoning the PII concept completely, since the raising number of re-identification attacks has shown that generally each and every combination can be used to identify an individual, for example, their movie preferences (see


Personal data
Anonymous information
If an individual is...
directly identifiable
indirectly identifiable
likely to be identifiable, as identifiability risk is insufficiently remote...
unlikely to be identifiable, as identifiability risk is sufficiently remote...
impossible to identify
...then the information is...
personal data
effectively anonymised
truly anonymous
...and data protection law applies.

...taking into account the **means reasonably likely to be used**, with consideration of the:

- data and its environment;
- context, scope and purposes of the processing; and
- technical and organisational measures applied.

With identifiability risk considered in terms of objective factors, including:

- motivation;
- competence needed;
- cost and time required;
- the available technologies; and
- legal gateways and likelihood of their use.

...and data protection law does **not** apply.
But keep things under review,
as appropriate.

Figure 2.2: The identifiability spectrum according to the ICO. Based on [ico22].

Section 2.8.4). Schwartz and Solove [SS11] disagree with the abandonment. In their opinion, the concept of PII plays a vital role in providing the boundaries of privacy law. They introduce an indentifiability continuum (the *PII 2.0 model*) which places information in a tripartite categorization of relating to an identified, identifiable, or non-identifiable person. Information about an *identified* person describes a singled-out individual, in other words, an ascertained identity. Information relates to an *identifiable* person when the identification is possible, but not significantly probable. Finally, *non-identifiable* information introduces only a remote risk for identification, when taking into account reasonably likely means for identification. A similar reasoning is provided by Polonetsky, Tene, and Finch [PTF16] who argue against the binary PII/non-PII approach and for an identifiability spectrum. Their introduced spectrum ranges from *explicitly personal* data, containing all unaltered information, over different categories of *pseudonymous* and *de-identified* data, to anonymous data.

2.3.2 Anonymization and De-Identification

The term *anonymization*, while commonly used in various fields, induces a lot of discussions in computer science and the legal community. Informally, everyone has a basic intuition of anonymization, somewhere around the lines of *transforming data about individuals in a way that it cannot be associated with these individuals anymore, so that the altered dataset can be published without harming the privacy of individuals*. However, when looking into the meanings and implicit assumptions in more detail, they sometimes differ to a large extent. Altman et al. [Alt+21] provide a high-level overview of different interpretations of *anonymization*. Amongst others anonymization can be understood as

- transforming data using specific techniques like aggregation, suppression, random swapping, and pseudonymization (see Sections 2.4 and 2.5),
- transforming data in a way that guarantees some specific property of the output, for example, k-anonymity (see Section 2.6.2),
- transforming data in a way that makes certain disclosure risks, such as identity disclosure or attribute disclosure (see Section 2.1), unlikely or impossible, or
- transforming data in a way that frees it from regulation, in other words, the regulation implicitly defines anonymization via its scope.

Additionally, as Ohm [Ohm09] argues, anonymization is often understood in an absolute way, that is, achieving perfect anonymity and protecting individuals from all privacy risks, while in practice and when taking the multitude of privacy breaches of "anonymized" datasets into account, it should be understood as effort to achieve this goal. Due to these inconsistencies, many people in law, computer science, and standardization organizations vote for abandoning the term altogether [Sta17; Gar14; Ohm09; RH16; Alt+21; Par14].

In this section, we give an overview of the term *anonymization* in computer science and law.

The EU Perspective

In the GDPR the term *anonymous* occurs only in Recital 26:

> The principles of data protection should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable. This Regulation does not therefore concern the processing of such anonymous information, including for statistical or research purposes.

Anonymization in the GDPR is therefore strongly based on the concept of identifiability and personal data. If data is not considered personal data in terms of identifiability (see Section 2.3.1), it is understood as anonymous data. The definition of personal data already includes an assessment of the likelihood of identifiability. If an individual is not identifiable in a dataset with the help of all means "reasonably likely" to be used, then the data is not considered personal data but anonymous data. Hence, a valid anonymization process is expected to output data for which identifiability is highly unlikely. But this also indicates that a remaining risk of identifiability is valid for anonymous data. Risk-freeness[8] is not required by the GDPR [DI20]. Furthermore, the GDPR does not prescribe any particular anonymization techniques – the sole requirement is given by the likeliness of identification [Par14].

The EU *Article 29 Data Protection Working Party* in their *Opinion 05/2014 on Anonymisation Techniques* provides some guidance for the practical implementation of this concept. They present several contextual factors to consider when choosing adequate anonymization techniques, amongst others the nature and sensitivity of the original data, sample size, availability of related public information, cost and required know-how for re-identification, and envisaged release of data to third parties. Furthermore, the Working Party assesses the suitability of different techniques for anonymization regarding their ability to prevent singling-out, linkability, and inference (as covered in Section 2.3.1).

There are some criticisms on several recommendations of the Working Party formulated by El Emam and Álvarez [EÁ14]. While the Working Party sees the requirement of deleting the original data for any transformed dataset to potentially qualify as anonymous, El Emam and Álvarez [EÁ14] argue for treating the data recipient as additional context factor when determining the identification risk. When a recipient is not in possession of the original data, it should be possible to consider the transformed dataset as anonymous with respect to this recipient. In their opinion the absolute deletion requirement contradicts the risk-based approach.

The working party mentions to treat *any* third party as a potential adversary when assessing the identification risk. El Emam and Álvarez demand a more precise wording here, as otherwise there would be no context-dependent assessment. They recommend to interpret this requirement as any third party that "has the same context as the data recipient and is a 'motivated intruder'" [EÁ14] – a concept provided by the ICO (see below).

[8] Anonymization resulting in data entailing no risk of identification is also referred to as *absolute anonymity* or *perfect anonymity*, in contrast to *factual anonymity* (also referred to as functional or *computational anonymity*) as demanded by the GDPR.

Another critique by El Emam and Álvarez relates to linkability of data records within the same database as a generally undesirable property of anonymization techniques. This would prohibit anonymous longitudinal data, without necessarily increasing the identification risk.

Recently, the ICO provided guidance on anonymization, pseudonymization and privacy enhancing technologies [ico22][9]. They follow the opinion of El Emam and Álvarez regarding the requirement of deleting the original data. The same data can represent personal data to one organization but anonymous data to another depending on the identification risk controlled by contextual factors of the data processing. They provide an example of pseudonymous data:

> An organisation applies a pseudonymization technique that divides personal data into two parts – a dataset that by itself does not identify individuals, and 'additional information' such as a key that enables re-identification. The organisation may refer to the first set as 'anonymous information'. This may indeed be the case in the hands of a third party that has no means reasonably likely to be used to re-identify individuals within that dataset. [ico22]

This also means that an unrestricted public release of anonymous data potentially requires more robust anonymization techniques than the non-public release to specific groups.

Another central argument in the guidance relates to the implicit risk-based approach taken by the GDPR when assessing whether data is considered anonymous or personal data. It is not required to take every hypothetical risk of identifiability into account but just what is *reasonably likely* with respect to the processing context. Instead, the goal of anonymization is to find the right balance between remaining risk and utility. We have already presented their spectrum of identifiability in Section 2.3.1. As a tool for assessing the identifiability, they provide the concept of the *motivated intruder test*. The idea is to evaluate whether a intruder, who is determined to identify individuals, is able to do so. They are expected to be reasonably competent, have access to appropriate resources, and use investigative techniques, but, on the other hand, do not have specialist technical knowledge, access to special equipment, and do not illegally gain data access. But the specifics of a potential intruder still should incorporate the context of the anonymization process, including the type and sensitivity of the data, the intruder motivation and potential knowledge, and the circumstances of the data release. Further details are provided in the guidance [ico22].

[9] We incorporate contents from a draft of the document that was open for comment during a consultation phase. The final guidance documents incorporating these comments have not yet been published at the time of writing this theses and are expected in spring 2023.

The US Perspective

Similar to the GDPR, in which anonymous data is characterized as non-personal data, the applicability of privacy laws in the USA[10] is based on the presence or absence of PII (see Section 2.3.1):

> Accordingly, organizations around the world have structured their internal and external privacy policies and practices around variations of PII – and its converse, de-identified data – locking themselves into a binary that does not accurately reflect how data are treated in practice. [PTF16]

A central term for these laws is *de-identification* – data can be de-identified so that it does not qualify as PII anymore. However, the specific meaning of this term differs a lot in the community. For example, Ohm [Ohm09] equates it with *release-and-forget* anonymization[11] and removal of PII, while Rubinstein and Hartzog [RH16] see it as an umbrella term for data transformation and data control methods aiming at the prevention of identity disclosure of individuals from data. Directly related to de-identification is the concept of *re-identification risk* – the risk which has to be assessed for qualifying data as (un-)successfully de-identified. The discussion around this concept includes similar considerations like the one related to identifiability in the GDPR. Polonetsky, Tene, and Finch [PTF16] provides some relevant open questions with respect to uncertainties in determining the re-identification risk and the success of de-identification, including:

- Is it required to really re-identify a particular record to show that the de-identification of a dataset was not successful? Or is an increased probability for re-identification already sufficient?
- How many records need to be re-identified to show that a dataset does not qualify as de-identified?
- Is the identification of a specific individual required or is singling out an individual from the dataset enough?
- How much confidence is required to qualify a dataset as de-identified or re-identified?

An example for a Unites States privacy law based on the concept of de-identification is the HIPAA dealing with privacy in the healthcare sector. § 164.514 contains the relevant details regarding de-identification. There are two possibilities to make data quantify as de-identified for the purposes of HIPAA: First, an expert can determine (and document) a small risk of identifying an individual by the anticipated data recipient, in which an expert is defined as a “person with appropriate knowledge of and experience with generally accepted statistical and scientific principles and methods for rendering information not individually identifiable”. Second, there is an explicit list of 18 attribute

[10] In comparison to the universal GDPR in EU, in the US privacy regulations are sector-specific [Nis+17]. For example, the Health Insurance Portability and Accountability Act (HIPAA) deals with privacy in the healthcare domain and the Family Educational Rights and Privacy Act (FERPA) with the education domain. An overview is provided by epic.org [epi].

[11] The term *release-and-forget anonymization* was coined by Ohm [Ohm09]. It refers to a anonymization method, in which data is modified to protect the individual’s privacy and then published publicly or to third parties without caring about anything regarding the data after publication. In other words, this describes the common method a layperson considers as anonymization.

types, including names, dates, and social security numbers, which need to be stripped or generalized for data to qualify as de-identified (known as the *Safe Harbor* method). Additionally, the processing party must "not have actual knowledge that the information could be used alone or in combination with other information to identify an individual who is a subject of the information."

Computer Science Perspective

There are conceptual differences between views regarding *anonymization* predominant in computer science and law. While, as we have seen, in law generally a risk-based approach to anonymization is taken, computer scientists tend to employ a worst-case approach [Yak11]. They aim to prevent all risks for all individuals in a dataset, against all adversaries, independent of the processing context. In comparison to legal concepts like personal data, identifiability, and risk, in computer science formal models of privacy are the main focus [CN20]. This is also indicated by the language employed. In computer science, the term *anonymization* is used less frequently. More often researchers speak about specific *privacy metrics*, *privacy models*, *privacy mechanisms*, or *data privacy definitions* and datasets or processes complying with the specific property. Examples for these include the syntactic privacy models (covered in Section 2.6) and formal mathematical models like DP (see Section 2.7).

2.3.3 Pseudonymization

Another concept for preserving the privacy of individuals in datasets that is often mentioned in the same breath as anonymization is pseudonymization. In this section we provide details on the concept of pseudonymization in law and computer science.

Law Perspective

Pseudonymization is a central concept in the GDPR. In the USA, the first privacy regulation to mention the concept is the California Consumer Privacy Act (CCPA), which bases its definition on the GDPR [LLP19]. Therefore, in this section we focus on the GDPR perspective to pseudonymization.

The GDPR defines pseudonymization as "the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information". For this purpose, typically one or more identifiers are replaced with random or derived pseudonyms in the dataset. The *additional information* (also referred to as *pseudonymization secret* [ENI22]) provides the link between pseudonyms and identifiers. The GDPR emphasises the importance of keeping this information separately from the data and protecting it against unwanted access using appropriate technical and organizational measures. Attention deserves the special status pseudonymization holds in the GDPR. For example, it is explicitly mentioned in Art. 25 as a possible technique for implementing the principle of *data protection by design*. This is unusual, since mentioning specific techniques in laws is rather the exception.

The definition includes an assessment of the possibility to link the data to a data subject. Without the additional information linking an individual with the respective pseudonym is expected to be hard, which expresses the assumption that pseudonymous data without access to the additional information in the GDPR sense must provide de-facto anonymity. However, the GDPR considers pseudonymous data still as personal data due to the possibility of re-identification. The Article 29 Working Party argues that the possibility of indirect identification always renders pseudonymous data personal data [Par14]. On the other hand, as mentioned in the example in Section 2.3.2, from the ICO's point of view the pseudonymous data might qualify as anonymous information with respect to parties without access to the additional information. El Emam and Álvarez [EÁ14] additionally criticize that the Article 29 Working Party sees linkability of pseudonymous data records in the same database as a disadvantage, while they consider this as one of the key benefits of pseudonymization.

Computer Science Perspective

Computer science cares more about technical means for pseudonymization than about legal implications depending on linkability. These techniques and further technical aspects about pseudonymity are covered in Section 2.4.

Pfitzmann and Hansen [PH10] define a pseudonym as "an identifier of a subject other than one of the subject's real name" and pseudonymity as using pseudonyms as identifiers. Furthermore they see pseudonymization just as the application of a mechanism, which in principle says nothing about the identifiability of the pseudonym holder and the potentially achieved privacy.

ISO Standard 20889 [Sta18] defines a *pseudonym* as a "unique identifier for a data principal to replace the commonly used identifier" and *pseudonymization* as a "de-identification technique that replaces an identifier (or identifiers) for a data principal with a pseudonym in order to hide the identity of that data principal". ISO Standard 25237 [Sta17] defines a *pseudonym* as a "personal identifier that is different from the normally used personal identifier and is used with pseudonymized data to provide dataset coherence linking all the information about a subject, without disclosing the real world person identity". In addition, the standard notes that pseudonyms usually do not allow the direct linkage to the normal personal identifier (making pseudonymous information "functionally anonymous"). It further defines *pseudonymization* as a "particular type of de-identification that both removes the association with a data subject and adds an association between a particular set of characteristics relating to the data subject and one or more pseudonyms".

In conclusion, pseudonymization in computer science focuses on using pseudonyms as identifiers which enable linking related data records. The (im-)possible identifiability of individuals is no prerequisite.

After having investigated the legal as well as technical interpretations of relevant terms, in the following sections we provide detailed information about technical measures, in particular pseudonymization in Section 2.4, de-identification techniques in Section 2.5, syntactic privacy models in Section 2.6, and semantic privacy models in Section 2.7.

2.4 Pseudonymity

In this section we cover relevant literature concerning pseudonymization. This includes pseudonymization scenarios (Section 2.4.1), which differ in the parties responsible for data processing and the pseudonymization itself, different pseudonym types and their connection to linkability (Section 2.4.2), generic attack techniques against pseudonymization (Section 2.4.3), and techniques for pseudonymization (Section 2.4.4).

2.4.1 Scenarios and Responsible Parties

When applying pseudonymization in a specific scenario, it is important to consider, amongst others, the goal of pseudonymization, the involved actors, who is responsible for the pseudonymization, and what utility is required from the pseudonymized data. The European Union Agency for Cybersecurity (ENISA) provides an overview of six different pseudonymization scenarios depending on the actors and goals of the pseudonymization process [ENI19]. In accordance with [ENI19], in the following the term *data controller* is used for the party which is responsible for the means of personal data processing, *data processor* for the party which performs the processing, and *pseudonymization entity* for the party which performs the pseudonymization.

1. **Pseudonymization for internal use**: The data controller collecting individual's data performs the pseudonymization of the data to reduce the risk for individuals caused by further internal processing, for example, in a different department of a corporation, or by security incidents.

2. **Processor involved in pseudonymization**: This scenario is similar to the first one with the only difference that an additional data processor is responsible for data collection on behalf of the data controller, while the controller still performs the pseudonymization.

3. **Sending pseudonymized data to a processor**: In this scenario the controller collects individual's data and performs the pseudonymization. The processor receives only pseudonymized data for further processing.

4. **Processor as pseudonymization entity**: Here, the data processor also takes the role of the pseudonymization entity and the data controller only receives pseudonymized data. This shifts the risk on the controller's side to the side of the processor. The controller is still able to re-identify data subjects through the data processor.

5. **Third party as pseudonymization entity**: In this scenario the data is collected and pseudonymized by a trusted third party (TTP). In contrast to the previous scenario the controller is not able to re-identify data subjects anymore[12].

6. **Data subject as pseudonymization entity**: In this scenario the data subject acts as the pseudonymization entity and creates their own pseudonym. Just like in the previous scenario, the controller has no means to re-identify a subject.

[12] To be more precise, they cannot revert the pseudonymization process by using the pseudonymization secret. Re-identification attacks like the ones covered in Sections 2.6.8 and 2.8 can still pose a threat.

Roßnagel and Scholz [RS00] provide a similar differentiation between pseudonym types depending on the party responsible for assigning the pseudonym to an individual. The first type are pseudonyms chosen by an individual themselves, such as usernames for online services. The next type are pseudonyms assigned by the party also processing the pseudonymized data. These pseudonyms protect the pseudonym holder only against other parties, for example, when data is published by the processing party. The final type are pseudonyms assigned by a trusted third party, organizationally separated from the data processing party. The data processing party potentially cannot link the pseudonymized data to the pseudonym holder without support from the trusted third party.

2.4.2 Pseudonymity and Linkability

One can differentiate between different types of pseudonyms according to their potential linkability (usage policy). Pfitzmann and Hansen [PH10] introduce a categorization focusing on the pseudonym holder with respect to communication relationships:

- **Person pseudonym**: A person pseudonym is a replacement for the pseudonym holders identity in many contexts, for example, their social security number.
- **Role pseudonym**: A role pseudonym is assigned to a pseudonym holder in a specific role, for example, a job-related pseudonym.
- **Relationship pseudonym**: A distinct relationship pseudonym is used for each communication partner.
- **Role-relationship pseudonym**: The combination of distinct pseudonyms according to the pseudonym holders role and the communication partner.
- **Transaction pseudonym**: Each transaction requires a new pseudonym unlinkable to any previously used pseudonym.

Even though most of these categories are based on communication relationships and are not directly applicable to data pseudonymization, the categorization highlights the relationship of pseudonym categories and the concept of (un-)linkability. They constitute a series of decreasing linkability. In the case of data pseudonymization and person pseudonyms, data records containing the same pseudonym can be linked to the same entity (even though linking the pseudonym to an individual might not be possible). In comparison, for transaction pseudonyms all data records belonging to one entity are assigned different pseudonyms and cannot be linked.

A more applicable categorization for data pseudonymity is given by the ENISA [ENI19]. The authors differentiate between *deterministic pseudonymization* (an identifier is always assigned to the same pseudonym), *document randomized pseudonymization* (an identifier is assigned to the same pseudonym just in a consistent scope), and *fully randomized pseudonymization* (an identifier is assigned to a different pseudonym each time). Similar considerations with respect to linkability to those of Pfitzmann and Hansen can also be made for this categorization. The linkability of pseudonymized data records for the same pseudonym holder decreases from deterministic pseudonyms (high linkability) over document randomized pseudonyms to fully randomized pseudonyms (low linkability).

2.4.3 Attacks on Pseudonymity

Using pseudonymization in a system requires thoughtful planning of adequate technical measures. It is especially necessary to think about potential weaknesses and possible attacks, which is the focus of this section. Note however, that we just cover weaknesses and attack techniques directly related to pseudonymization. Other, more sophisticated attacks on sensitive data with the goal of re-identifying individuals which do not attack the pseudonymization process will be covered in Section 2.6.8.

One can differentiate between three attack goals [ENI19]:

- **Attacks to disclose the pseudonymization secret**: If an adversary learns the pseudonymization secret (mapping table, used cryptographic key, etc.), they disclose the identity of each and every pseudonym holder.
- **Re-identification attacks**: The goal of a re-identification attack is to link one or more pseudonyms to their respective pseudonym holders. While a successful disclosure of the pseudonymization secret is the most severe kind of re-identification attack, a re-identification attack can also just target subsets of or even individual pseudonym holders.
- **Discrimination attacks**: This goal describes attacks which identify properties of a pseudonym holder without directly identifying the individual behind the pseudonym. Gathered information may already lead to discrimination or may be used as a basis for further re-identification attacks.

Furthermore there are three generic attack techniques against pseudonymization methods [ENI19]:

- **Brute-force attacks** (a.k.a. exhaustive search): If the adversary has oracle-like access to the pseudonymization function f_P (that is, they can compute or query pseudonyms for arbitrary identifiers), to find the identity of a pseudonym holder for a pseudonym P_i they can iterate over all possible identifiers $I_j \in I_1, \ldots, I_n$ until $f_P(I_j) = P_i$. The feasibility of this attack depends on the size of the identifier domain. If the pseudonymization secret (for example, the cryptographic key used in a message authentication code (MAC) calculation) is unknown to the attacker, they can try to brute-force this secret as well. Therefore it is important to choose large enough secret domains to prevent this attack, if possible, or rely on techniques which do not provide adversaries with access to the pseudonymization function f_P.
- **Dictionary attacks**: These attacks are special cases of brute-force attacks in which pairs of identity and pseudonym $(I_i, P_i = f_P(I_i))$ are precomputed and stored by an attacker so that the exhaustive search is replaced by a simple dictionary lookup. They allow to replace the time required for the brute-force attack with large amounts of memory for storing the precomputed dictionary.
- **Guesswork**: By employing background knowledge about pseudonym holders or the pseudonymization function (for example, in the form of statistical characteristics of identifiers), an adversary might be able to guess the identity of a pseudonym holder with higher probability than what would be achievable by exhaustive search.

Table 2.2: Examples for pseudonymization via mapping tables.

Identifier	Pseudonym
I_1	1
I_2	2
I_3	3
...	

(a) Counter

Identifier	Pseudonym
I_1	c732ad5e
I_2	be9eb24d
I_3	416bdfa8
...	

(b) Random

When choosing an adequate pseudonymization mechanism, one has to keep these attack techniques in mind.

2.4.4 Techniques

In the following sections we describe simple mechanisms which can be used for pseudonymization. While we look at the process for single identifiers, it can easily be extended to multiple identifiers, for example, by concatenating these identifiers. The covered mechanisms present basic means for creating pseudonyms. Sophisticated cryptographic primitives allow for pseudonymization processes which can fulfill special requirements. Several examples are presented by ENISA [ENI21], for example, using a zero-knowledge proof (ZKP) to allow a subject to proof the ownership of a pseudonym without revealing their identity.

Mapping Table

A simple variant for pseudonymization is the utilization of a mapping table as depicted in Table 2.2. The approach is also referred to as *Tokenization* [ENI18]. For each occurring identifier I a pseudonym P_I is created and the pair is stored in the mapping table, which serves as the pseudonymization secret and can also be used for re-identification of pseudonym holders. The pseudonym can be generated by a simple incrementing counter (Table 2.2a) or completely random (Table 2.2b). While the counter method is simple to implement, depending on the context the order of pseudonyms can leak information, which might be used by an adversary for re-identification attacks. When generating random pseudonyms, collisions should be prevented by checking for already existing pseudonyms (this might be unnecessary, if large amounts of random bytes are used). Since there is no direct connection between identifier and pseudonym, an adversary has no possibility for brute-force or dictionary attacks. One possible disadvantage of this approach is that it requires larger amounts of storage since the full mapping table has to be stored in comparison to a single cryptographic key in other approaches.

Hashing

A simple, yet insecure way of deriving pseudonyms from identifiers is the usage of a cryptographic hash function h. For an identifier I we use the hash value $h(I)$ as the pseudonym. Since hash functions are deterministic, an identifier is always translated to the same pseudonym. In theory, collisions in the hash output can occur because of the fixed output domain size of hash functions. This would lead to the assignment of the same pseudonym to different identifiers. But due to the large output domain of established hash functions like Secure Hash Algorithm (SHA)-3 [nis15; Ber+11], this is likely to be not of relevance in practice.

Due to the irreversibility of hash functions, an adversary is not able to directly compute the identifier from the pseudonym. However, as covered in Section 2.4.3, one possible attack against pseudonymity is based on exhaustive search. Since simple cryptographic hashing does not employ a pseudonymization secret, an adversary can perform a brute-force attack using all potential identifiers. This unsuitability of hashing for the generation of pseudonyms has already been shown multiple times[Mar+18; Dem+18]. For this reason, using hash functions for the pseudonymization of identifiers is generally not recommended [ENI18].

(Hash-based) Message Authentication Codes

The weakness of a hash-based pseudonymization function can be fixed by introducing a cryptographic key as pseudonymization secret in the process. ehe cryptographic primitive for this purpose is the MAC and a common instantiation is hash-based message authentication code (HMAC) [BCK96; KBC97] in combination with a secure hash function like SHA-3 [nis15; Ber+11]. For the pseudonymization process, a key k is chosen randomly and used in the process for generating pseudonyms. For an identifier I we use $hmac_k(I)$ as the pseudonym. This construction (for a fixed key k) is still deterministic, so an identifier is always assigned to the same pseudonym. HMAC values, like hash values, are irreversible, so that even with knowledge of the used key k the identifier can not be computed directly from the pseudonym. This principle allows for linking data belonging to one subject while not requiring to store their identifiers – a possibility to follow the principle of *data minimization* in the pseudonymization process [ENI18].

The cryptographic key used in the construction is kept secret, so that adversaries are not able to perform brute-force or dictionary attacks on identifiers (even for small finite domains) without the knowledge of this key. While the key length in the HMAC construction is not fixed, currently common choices of at least 256 bit are large enough to prevent brute-force attacks against the used key.

Symmetric encryption

Another possibility for the generation of pseudonyms is the usage of symmetric encryption. For a randomly chosen symmetric key k, the pseudonym for an identifier I is computed as the symmetric encryption $E_k(I)$. The symmetric key serves as the pseudonymization secret in this method. A common choice for the encryption algorithm would be Advanced Encryption Standard (AES) with an appropriate sized key. As with the MAC-based solution the key is kept private which prevents adversaries from performing brute-force or dictionary attacks. But in contrast to the MAC-based solution, the pseudonymization process is reversible when a party is in possession of the symmetric key by simply decrypting the pseudonym.

Special considerations must be put into the choice of the operation mode [ENI19]. Depending on the size of the identifier domain, it might be possible to use the (deterministic) electronic codebook (ECB) mode and just produce a single encrypted block for identifier domains with values smaller than the AES block size of 128 bit. For identifier domains with larger values, the choice of an adequate operation mode can prove tricky. Choosing the ECB mode can leak information about the identifiers, such as common prefixes or equal block-sized parts, while choosing secure modes like Galois/counter mode (GCM), which use a random initialization vector (IV), causes indeterministic behavior and prevents the assignment of the same pseudonym to an identifier for multiple encryptions. Depending on the context and identifier domain, an alternative approach for large domain values can be the compression of values into single blocks to be encrypted [ENI19].

2.4.5 Concluding Remarks

As we have seen, pseudonymization reduces the linkability of a data record with an individual by replacing identifiers with a pseudonym. But it does so, while still allowing for the re-identification of individuals with the help of a pseudonymization secret, if desired. Additionally, pseudonymization can enable the linking of multiple data records belonging to the same individual – even across multiple datasets. The technique plays an important role as a privacy-protection technique in the GDPR (see Section 2.3.3). But it comes with no guarantees regarding re-identification risks for adversaries not in possession of the pseudonymization secret. Designing adequate pseudonymization solutions therefore should follow a risk-based approach, considering the purpose and context of data processing as well as necessary utility levels [ENI18].

2.5 De-Identification Techniques

De-identification techniques (also referred to as *(data) masking methods* and *disclosure control/protection methods*) describe a set of techniques which aim at modifying datasets in a way which allows a publication of the dataset without putting the data of individuals at risk. As we have argued in Section 2.3.2 the definition of de-identification comprises a large spectrum of meanings. Therefore in this chapter we focus on techniques which do not qualify as pseudonymization and syntactic or semantic privacy models. Nonetheless, some of the covered techniques such as generalization (see Section 2.5.1) or perturbation (see Section 2.5.6) play an important role as building blocks for algorithms achieving syntactic or semantic privacy.

Most of the early results have been provided by the statistics community, where the terms *statistical disclosure control* or *statistical disclosure limitation* are used to describe

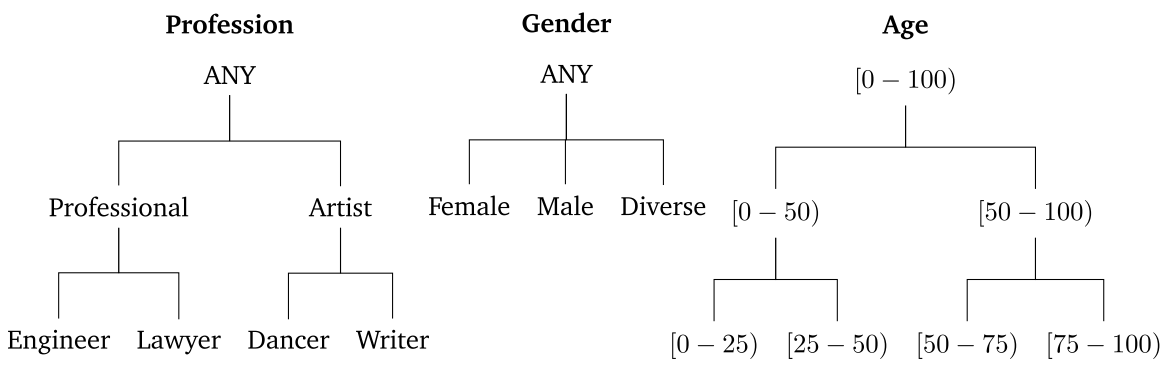


Figure 2.3: Example value generalization hierarchies. Figure based on [Fun+10a].

the study of de-identification techniques and their impact on statistical data [Dwo06]. One can find multiple reviews of results from this field in the literature [MH11; Hun+12; WD12; Tem17]. Additionally, there is a large amount of reviews from the field of computer science [Dom08; Fun+10a; El 13; Sta18].

In the following, we provide a discussion of methods that are frequently included in the aforementioned surveys. We refrain from going into details about the statistical influence of these methods on re-identification risk or data utility, because these methods are rarely used in isolation in practice today. Additionally and unlike syntactic and semantic privacy models, these assessments are generally statistical in nature and provide no provable guarantees. Readers seeking a more comprehensive understanding of these methods are encouraged to refer directly to the cited surveys.

2.5.1 Generalization

Generalization describes techniques that reduce the granularity of attribute values according to a given taxonomy. A taxonomy is often provided in the form of so-called *value generalization hierarchies* [Sam01] (also referred to as *taxonomy trees* [Fun+10b]), in which each leaf node represents a specific attribute value, each parent node is a generalization of its child nodes, and the root node represents the most general value, which embodies all possible values. An example for attributes *profession*, *gender*, and *age* is displayed in Figure 2.3. An alternative is the partition-based approach where each attribute is considered as an ordered set and generalization relationships are defined by partitioning the set into disjoint subsets. We can differentiate between different styles of generalization [LDR05]. An overview is provided in Figure 2.4.

Local recoding (also referred to as *cell generalization* [Fun+10b]) allows generalization of individual data records. For a generalization $p \rightarrow \{c_1, c_2\}$ of attribute a, local recoding allows that some records in the dataset keep the more general attribute value p, while others are generalized to attribute values c_1 and c_2. While this method can potentially provide the least distorted data, this comes at the cost of complicated data analysis since different records potentially contain completely different generalization levels of the same attribute [WD12]. For example, it is unclear how a data analyst would utilize a dataset containing records with *ANY* gender as well as *male*, *female*, and *diverse* records.

Global recoding describes a generalization approach in which all data records share the same generalization level for each distinct attribute value. Global recoding can be differentiated further into *single-dimension* or *multi-dimension recoding*.

Single-dimension recoding describes generalization approaches in which each dimension is considered for itself. Each data record containing value c for attribute a is generalized so that c is generalized to value p, where p is equal to c or represents a more general value, that is, a value on the path from c to the root node in the generalization hierarchy. For example, when considering the gender attribute in a dataset (see Figure 2.3) for each female subject their data record either keeps the attribute value *female* or is generalized to the value *ANY*. There are several variants of single-dimension recoding: *Full-domain recoding* requires all data records to share attribute values from the same level of the generalization hierarchy. As an example, for the *profession* generalization hierarchy in Figure 2.3 all data records would contain attribute values of the second level (*professional* or *artist*) *or* of the third level (*engineer*, *lawyer*, *dancer*, or *writer*). *Full-subtree recoding* requires that if a single value is generalized to p, all attribute values in the generalization hierarchy subtree under p must be generalized to p. Again referring to the *profession* attribute from Figure 2.3, if *lawyer* data records are generalized to *professional* this requires *lawyers* to be generalized as well, while *dancers* can remain specialized. *Unrestricted recoding* just requires valid generalizations but imposes no further restrictions on the generalization.

Multi-dimension recoding, in opposition to single-dimension recoding, looks at multiple attributes at once. In this approach, global recoding means that not each single attribute but only a combination of attributes must be generalized in the same way for all data records, which allows for more flexible generalizations. Multi-dimension recoding can be divided in *full-subgraph* and *unrestricted* recoding. *Full-subgraph recoding*, parallel to full-subtree recoding in the single-dimensional case, means that if a combination of attributes is mapped to some general attributes, all attribute values in the subtrees of the respective (single-dimensional) generalization hierarchies must be mapped to the more general ones. An example based on Figure 2.3 and the attributes *profession* and *gender*: if $\langle engineer, female\rangle$ is recoded to $\langle professional, ANY\rangle$, all engineers *and* lawyers of any gender must be recoded to $\langle professional, ANY\rangle$ as well. This restriction does not apply to *unrestricted* recoding. For example, unrestricted recoding allows to generalize $\langle engineer, female\rangle$ to $\langle engineer, ANY\rangle$ and $\langle lawyer, female\rangle$ to $\langle professional, female\rangle$. This example also shows the flexibility in comparison to single-dimension recoding, as the given recoding is not possible in single-dimension recoding.

In addition to these general approaches there are also more specific techniques, which can be seen as generalization techniques [Sta18]. One example is *rounding* values either in the classical mathematical way or probabilistically based on the distance of the attribute value to the nearest rounding base. For example, if the value 3 were to be rounded to the nearest 10, there was a 30 % chance it would become 0 and a 70 % chance it would become 10. Another example is *top-bottom-coding* (also referred to as *clamping*) where all attribute values above or below some upper and lower thresholds would be replaced with these thresholds. This is an approach to protect outliers in the attribute values.

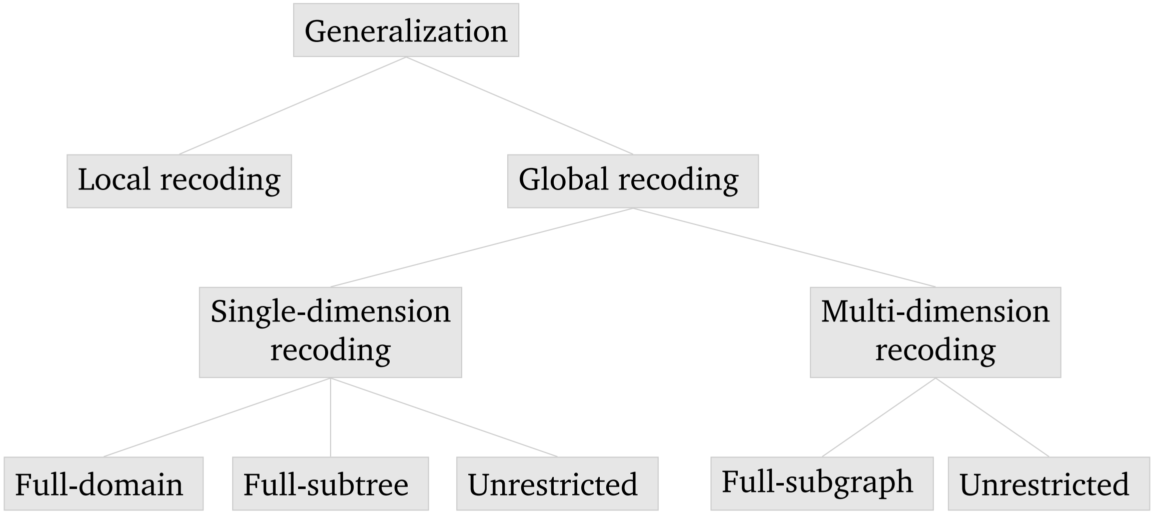


Figure 2.4: Taxonomy of generalization approaches.

2.5.2 Suppression

Suppression describes the complete removal of some feature in the dataset. It can be used if an attribute or attribute value is too sensitive for publishing or not required for the dataset (*data minimization*), or to remove outliers from the dataset. Practically this can also be achieved by replacing values with a special value indicating the suppression of this value, for example, if not all values in a table column should be removed. There are different types of suppression [Fun+10a; El 13]:

- *Record suppression* (also referred to as *case-wise deletion*) removes full data records from the dataset.
- *Attribute suppression* (or *quasi-identifier suppression*) removes specific attributes from all records of the dataset.
- *Value suppression* removes specific attribute values from the dataset.
- *Cell suppression* (or *local suppression*) removes just some instances of attribute values from the dataset.

2.5.3 Swapping

Data swapping (also referred to as *shuffling* or *permutation* [Sta18]) describes the interchanging of sensitive attribute values of records in the dataset [WD12]. While swapping removes the relationship between the sensitive attribute value and the remaining data record, it preserves the distribution of the sensitive attribute within the dataset for statistical analysis.

A special form of swapping is *rank swapping* [Fun+10a]. For this technique the sensitive values for attribute A are ordered and a value $x \in A$ is interchanged with another value $y \in A$ which is randomly chosen in the neighborhood of x with respect to the ordering of A. The neighborhood can be determined, for example, as a percentage of $|A|$.

Table 2.3: The two tables for an anatomized dataset.

Name	Sex	DOB	Postcode	Group
Alice	f	19.02.1978	54321	g_1
Bob	m	04.09.1983	54328	g_1
Carol	f	07.03.1979	54321	g_2
Dave	m	23.09.1975	54319	g_2
Eve	f	12.11.1978	54373	g_2

(a) The quasi-identifier (QID) table.

Group	Disease	Count
g_1	Flu	1
g_1	HIV	1
g_2	Flu	2
g_2	PAD	1

(b) The table for sensitive attributes.

Further considerations and results about swapping are provided by Matthews and Harel [MH11] and Domingo-Ferrer [Dom08].

2.5.4 Sampling

Sampling describes the publishing of a subset of the whole dataset, so that an individual can be linked to a record in the dataset only with some degree of uncertainty. This approach might work for categorical microdata, but can fail for continuous microdata when attributes are published unaltered since perfectly matching continuous values increase the probability of matching records [Dom08].

Rocher, Hendrickx, and Montjoye [RHM19] provide a way to estimate the likelihood of a correct re-identification in incomplete datasets. They show that the population uniqueness of individuals and with that the risk of correct re-identification increases heavily with the number of attributes in the dataset. Their result challenges the assumption that sampling a subset of individuals before publishing a dataset decreases the disclosure risks for individuals.

2.5.5 Slicing and Anatomization

Slicing is performed by separating the attributes of a dataset into groups and publishing each group separately (and in randomized order) [MH11]. This approach reduces the risk of re-identification attacks by limiting the attribute values related to on individual. For the same reason, it drastically reduces the data utility for analysis in which relationships between sliced attributes play a vital role.

A related technique is *anatomization* (also referred to as *bucketization*). Xiao and Tao [XT06] present the concept of *anatomization* and an anatomization-based algorithm which fulfills l-diversity (see Section 2.6.3). In comparison to generalization-based approaches (cf. Section 2.6.5) anatomization does not alter attribute values. Instead, for some attribute to be protected it creates groups of data records and just publishes aggregated counts for this sensitive attributes for each group. In this way the direct linkage of individual and sensitive attribute is prevented. An example is shown in Table 2.3.

Xiao and Tao show that this approach allows for more accurate query results in comparison to generalization-based methods. On the other hand, the authors admit that by releasing the unaltered attribute values anatomization may have an increased probability of re-identification in comparison to generalization.

2.5.6 Perturbation

Perturbation (also referred to as *noise addition*) describes an approach for numerical sensitive values. The basic idea is simple: Replace a sensitive value s with a perturbed version $s + r$ where r is random noise drawn from an appropriate distribution, such as the normal distribution with a mean $\mu = 0$ and a variance depending on the desired privacy protection. While this approach can reduce the data utility of a unique data record, it can preserve statistical properties of the dataset depending on the distribution the noise is drawn from. The additional gained privacy might, however, be less than expected due to the potential correlation of attributes [Kar+03]. But perturbation is a central concept for employing DP introduced in Section 2.7.

2.5.7 Post Randomization Method

The Post Randomization Method (PRAM) [GKW98] describes the application of the randomized response technique (see Section 2.7.3), not during a study but for an already existing dataset. For a sensitive attribute with two possible values the real value in a record is published with some probability p and its opposite value with probability $p - 1$. This provides *plausible deniability* for the sensitive value of each record. PRAM extends this simple mechanism to continuous variables and different probability distributions and in comparison to randomized response can depend on the real attribute value. This technique, while perturbing single data records, can preserve the option of performing statistical analysis.

2.5.8 Microaggregation

Microaggregation describes a technique in which the dataset is partitioned into groups and for each of these groups and some continuous attribute the record values are replaced with the average value within the group. While this approach alters individual attribute values, it preserves the sum and average of the attribute with respect to the whole dataset. The technique can be divided into *univariate* and *multivariate* microaggregation [WD12]. In the univariate case a single attribute is averaged in each group, while in the multivariate case this is performed for multiple attributes. Central decisions for the application of microaggregation deciding about data utility and privacy are the size of groups (fixed or variable) and how to partition the dataset into these groups, for example, completely random or based on similarity of data records [Dom08].

2.5.9 Synthetic Data

Instead of releasing the (potentially masked) real dataset, another approach, initially proposed by Rubin [Rub93], is to publish a synthetically generated dataset so that real data records of individuals are not published at all. The idea in *synthetic data generation* is to create a statistical model from the real dataset, that preserves statistical relationships of the dataset, and then to sample complete data records (*fully synthetic data*) or sensitive attribute values (*partially synthetic data*) from this model [MH11]. Specific techniques include statistical models, such as *Bayesian networks* and *hidden Markov models*, and non-parametric models, for example, *generative adversarial networks (GANs)* and *variational auto encoders* [SOT22].

Domingo-Ferrer [Dom08] points out that the circumvention of the re-identification problem might be less clear than apparent at the first glance. For example, it is possible that data records are generated which are equal to records from the real dataset by chance. Furthermore, the data utility can be limited since syntactic data provides just the statistical properties explicitly captured by the generating model. In this sense, it might be more reasonable to directly publish the relevant statistics. With regard to this problem, Willenborg and De Waal [WD12] mention the difficulty to control for all possible analyses data analysts might want to perform, for example, to provide enough statistical relationships for multiple subpopulations in the dataset.

Recent results of Stadler, Oprisanu, and Troncoso [SOT22] show that synthetic data generation is subject to the same privacy-utility tradeoff (see Section 2.2) as other traditional de-identification techniques: Either the synthetically generated data suffers from poor data utility or is not able to withstand inference attacks. Furthermore, they find that it is harder to assess this tradeoff for synthetic data generation. The authors conclude that “synthetic data is far from the holy grail of privacy-preserving data publishing”. A potential reconstruction attack against aggregate statistics and synthetic microdata is covered in Section 2.8.12.

Earlier results about synthetic data generation are reviewed by Domingo-Ferrer [Dom08]. Several surveys were published recently, amongst others focussing on privacy [Jor+22], on GANs [FV22], and on the health [Her+22] and finance sector [Ass+21]. Some results which include DP into synthetic data generation are covered in Section 2.7.6.

2.5.10 Concluding Remarks

In practice, it has been shown that using simple de-identification measures, such as stripping identifiers and generalizing some attributes, often leaves datasets vulnerable to re-identification attacks (see Section 2.8). Additionally the strict separation of attributes according to their potential for identifying individuals is not tenable, as the identification of individuals based on watched movies indicates exemplarily (see Section 2.8.4). These attacks led Ohm [Ohm09] to propose the “failure of anonymization”. Cohen [Coh22] mentions that these measures lack three important properties: they are not composable, not robust against post-processing, and they rely on assumptions with respect to data distributions. Narayanan and Shmatikov see de-identification only as an option to prevent easy data access by curious insiders and “to keep honest people honest” [NS19], for example, as an internal control mechanism to reduce the risk of employees peeking at data records of specific individuals. However, advances in re-identification science and more publicly available data about the population drastically decrease the effectiveness of these measures in their opinion. In general, it is difficult to assess the privacy impact of simple de-identification techniques, even if they give a feeling of protection at an intuitive level.

2.6 Syntactic Privacy Models

Sweeney demonstrated that using simple de-identification mechanisms like stripping identifiers are not enough to prevent identity disclosure (see Section 2.8.2). This lead to the emergence of syntactic privacy models which allow to check specific privacy guarantees based on the anonymized dataset.

In this section, we cover the theory behind syntactic privacy models and present their most well-known representatives k-anonymity [SS98b; SS98a; Sam01; Swe02], l-diversity [Mac+06], and t-closeness [LLV10]. Furthermore, we provide details about common algorithms to achieve syntactic privacy for a dataset and introduce a limitation of syntactic privacy models – the curse of dimensionality. Finally, we introduce several attacks on syntactic privacy models, which indicate their inherent weaknesses to specific threats.

We will use the following notations in this section, which mostly follow Machanavajjhala et al. [Mac+06]. A dataset (or table) T contains n data records (or rows) $t_1, \ldots, t_n$ and each record represents data related to one individual. A record consists of values (or cells) $t = \{a_1, \ldots, a_m\}$ for attributes (or columns) $\mathcal{A}_1, \ldots, \mathcal{A}_m$ with respective domains. Let $t[\mathcal{A}_i]$ denote the value of attribute $\mathcal{A}_i$ for record t and $t[\mathcal{C}]$ with $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_p\}$ the tuple $(t[\mathcal{C}_i], \ldots, t[\mathcal{C}_p])$ (a projection of t onto the attributes in $\mathcal{C}$). Furthermore, we denote the set of QIDs (see Section 2.6.1) with $\mathcal{QID}$ and the set of sensitive attributes with $\mathcal{SA}$.

2.6.1 Attribute Types

In the context of syntactic privacy models, data attributes are classified into four categories [MH22]:

- **Identifying attributes** (sometimes also referred to as *directly* identifying attributes): These are attributes which can be used to uniquely identify an individual, such as the name or the social security number. There exists some public or private database potentially accessible by an adversary through which the attribute can directly and uniquely be linked to the individual.
- **QIDs** [Dal86]: These are attributes[13] which, for themselves, are not enough to uniquely identify an individual, but can be used in combination with auxiliary information to identify individuals. Examples are the age, postal code, and gender of an individual.

[13] To prevent misunderstandings, we want to point out that in some publications the term *quasi-identifier* already refers to the full set of attributes instead of a single attribute [Coh22].

- **Sensitive attributes**: These describe the information an individual wants to keep private, such as their income or health information.

- **Nonsensitive attributes**: These are attributes which do not fall into any of the previous categories. It is hard to come up with a universal example for nonsensitive attributes, because there might nearly always be a scenario where suitable auxiliary information can be used to deduce something from an attribute.

Classifying the attributes of a dataset according to these categories is a crucial step for protecting the privacy of individuals in a anonymized dataset. El Emam [El 13] and Fung et al. [Fun+10a] provide some guidelines on how to categorize attributes according to these attribute types. These include the idea to categorize attributes an adversary might obtain from external sources (potential adversarial background knowledge) as QIDs – even though it is by no means obvious how to determine this. Narayanan and Shmatikov [NS10], on the other hand, argue that in principle each and every information distinguishing individuals can be used for re-identification.

2.6.2 k-Anonymity

Samarati and Sweeney [SS98b; SS98a; Sam01; Swe02] introduced the first syntactic privacy model, called *k-anonymity*. The basic idea is to hide the identity of an individual and therefore the connection to their sensitive attribute(s) in a group of other individuals. A dataset is said to be k-anonymous if the data of an individual cannot be distinguished from at least $k-1$ other data records with respect to chosen QIDs. The following definition is based on Machanavajjhala et al. [Mac+06].

Definition 2.6.1. More formally, a dataset T satisfies k-anonymity if for every record t at least $k-1$ other records $t_{i_1}, \ldots, t_{i_{k-1}}$ exist with $t[\mathcal{QID}] = t_{i_1}[\mathcal{QID}] = \cdots = t_{i_{k-1}}[\mathcal{QID}]$. The set of records sharing the same QID values is referred to as *equivalence class* (sometimes also *equivalence group* [Moh+10]).

As an example, Table 2.4 shows a dataset that includes patient identities, some demographic information, and the diseases they suffer from. Table 2.5 depicts a 2-anonymous version of the original dataset. The identifying attribute, in this case the name, was completely removed and the sensitive attribute, in this case the disease, was left untouched. The remaining attributes sex, date of birth, and postcode were categorized as QIDs and were generalized in a way that the two resulting equivalence classes contain at least $k = 2$ records.

This property prevents an adversary who knows the QID values of an individual to single out a data record and to deduce the sensitive attribute of the individual. Therefore it protects against identity disclosure (see Section 2.1). However, as Machanavajjhala et al. [Mac+06] have shown, k-anonymity is vulnerable to *background knowledge* and *homogeneity attacks* (see Section 2.6.8) because k-anonymity does not take the distribution of sensitive values in equivalence classes into consideration.

Table 2.4: The example dataset for k-anonymity in its unmodified form.

Identifying Attribute	QIDs			Sensitive Attribute
Name	**Sex**	**DOB**	**Postcode**	**Disease**
Alice	f	19.02.1978	54321	HIV
Bob	m	04.09.1983	54328	Stroke
Carol	f	07.03.1979	54321	HIV
Dave	m	23.09.1975	54319	PAD
Eve	f	12.11.1978	54373	Breast cancer
Frank	m	12.11.1981	54325	Flu

Table 2.5: A possible 2-anonymous variant of the original dataset with three equivalence classes.

QIDs			Sensitive Attribute
Sex	**YOB**	**Postcode**	**Disease**
f	1978–1979	54321	HIV
m	1981–1983	5432*	Stroke
f	1978–1979	54321	HIV
*	1975–1978	543**	PAD
*	1975–1978	543**	Breast cancer
m	1981–1983	5432*	Flu

2.6.3 l-Diversity

To overcome the vulnerability of k-anonymity to background knowledge and homogeneity attacks (see Section 2.6.8) Machanavajjhala et al. [Mac+06] present l-diversity. Contrary to k-anonymity this privacy model tries to deal with attribute disclosure (see Section 2.1). The main idea of l-diversity is to put requirements on the distribution of the sensitive attributes in equivalence classes, so that an adversary cannot deduce the specific sensitive attribute of an individual in the equivalence class. The following definition is provided by Machanavajjhala et al. [Mac+06].

Definition 2.6.2. An equivalence class[14] is l-diverse if it contains at least l "well-represented" values for the sensitive attribute. A dataset T is l-diverse if every equivalence class is l-diverse.

This definition requires an adversary to possess at least $l-1$ pieces of background information to eliminate $l-1$ possible sensitive values and disclose the sensitive attribute of an individual. Machanavajjhala et al. provide two practical methods to instantiate the term "well-represented", namely *Entropy l-diversity* and *Recursive* (c,l)*-diversity* (and further variants of this method).

Entropy l-diversity, as the name suggests, uses the (Shannon) entropy of the sensitive attribute distribution in an equivalence class as a measure for a sufficiently diverse value distribution. The method requires the entropy to be at least $\log l$ in all equivalence classes. For an equivalence class q^* it is required that

$$-\sum_{s\in S} p(q^*,s)\log p(q^*,s) \geq \log l,$$

where $p(q^*,s)$ denotes the fraction of sensitive value s in equivalence class q^*.

Recursive (c,l)-diversity is based on the idea that no sensitive value should appear too frequently or too rarely in an equivalence class. Formally, let r_i denote the count for the i^{th} most frequent sensitive value in an equivalence class. The equivalence class is said to be (c,l)-diverse if $r_1 < c(r_l + r_{l+1} + \cdots + r_m j)$ for a given constant c. Machanavajjhala et al. provide further variants of this definition which allow higher frequencies for specific sensitive values or deal with prohibited negative disclosure for specific sensitive values.

Clifton and Tassa [CT13] point out, that in practice often a simpler method is used: An equivalence class fulfills l-diversity if the frequency of each sensitive value does not exceed $\frac{1}{l}$. Another simple, but weaker variant of l-diversity (sometimes referred to as *distinct* l-diversity [LLV07]) often found in practice requires that each equivalence class contains at least l distinct sensitive values. This variant was also independently introduced as *p-sensitive k-anonymity* by Truta and Vinay [TV06].

Table 2.6 shows an example of a distinct 2-diverse variant of the dataset depicted in Table 2.4. In contrast to the 2-anonymous variant presented in Table 2.5, which is

[14] Machanavajjhala et al. speak of q^*-blocks here.

Table 2.6: A possible distinct 2-diverse variant of the original dataset with two equivalence classes.

QIDs			Sensitive Attribute
Sex	**YOB**	**Postcode**	**Disease**
f	1978	543**	HIV
m	1981–1983	5432*	Stroke
*	1975–1979	543**	HIV
*	1975–1979	543**	PAD
f	1978	543**	Breast cancer
m	1981–1983	5432*	Flu

vulnerable to the homogeneity attack (see Section 2.6.8), the 2-diverse dataset contains no equivalence class which violates l-diversity.

Li, Li, and Venkatasubramanian [LLV07] show that l-diversity is vulnerable to *skewness* and *similarity* attacks (see Section 2.6.8), because the distribution of sensitive attribute values in a l-diverse equivalence class can still differ from the distribution in the population by a large amount.

2.6.4 t-Closeness

Therefore, Li, Li, and Venkatasubramanian [LLV07] propose another privacy model called t-closeness to tackle weaknesses of l-diversity (see Section 2.6.8). The basic idea is to bound the distance between the distribution of sensitive attributes in an equivalence class and in the full dataset by a threshold of t. This limits the individual-specific information an adversary can learn from an equivalence class. The following definition is provided by Li, Li, and Venkatasubramanian [LLV07].

Definition 2.6.3. Let $\mathbf{P}$ denote the dataset-wide distribution and $\mathbf{Q}$ the distribution in an equivalence class for a sensitive attribute and $D[\mathbf{P}, \mathbf{Q}]$ a distance measure between distributions $\mathbf{P}$ and $\mathbf{Q}$. The equivalence class fulfills t-closeness if $D[\mathbf{P}, \mathbf{Q}] \leq t$. A table T fulfills t-closeness if all equivalence classes fulfill t-closeness.

It remains to look at how to measure the distance between given distributions. Li, Li, and Venkatasubramanian analyze different distance measures regarding their suitability for t-closeness. They identify the so-called earth mover's distance (EMD) [RTG00] as a good fit since it implicitly incorporates some semantic meaning for the sensitive attributes. They further provide ideas on how to compute the EMD for numerical and categorical attributes. From the EMD properties follows the range of possible values for the parameter t: $0 \leq t \leq 1$.

A small t value allows less deviations of the two distributions and leads to less information an adversary can achieve from the equivalence class. However, as Frikken and

Zhang [FZ08] state, since there is no clear relationship between t and the information gain of an adversary, it is not-trivial how to choose t to prevent specific types of disclosure. Furthermore, a choice for t can have varying implications for privacy depending on the scenario. Finally, to avoid some specific types of disclosure risks, t has to be chosen so small that the data utility decreases significantly.

There are some results about relations between t-closeness and DP [Eke+22; DS15], which imply that these two concepts might be more related than generally assumed under certain circumstances.

2.6.5 Algorithms

In this section we cover algorithms for syntactic privacy models. Generally, these algorithms use de-identification techniques, most often generalization and suppression (covered in Section 2.5), to transform datasets so that they fulfill the privacy condition of a syntactic privacy model. It has been shown that finding optimal[15] solutions for syntactic privacy models is NP-hard [MW04; Agg+05; LDR06; XYT10], so existing algorithms use heuristic approaches to achieve their goal.

In the remainder of this section we present two well-known algorithms in more detail: *Incognito* [LDR05] and *Mondrian* [LDR06]. There is a large number of further algorithms, such as *Datafly* [Swe98], *TDS* [FWY05], *Hilb* [Ghi+07], and *OLA* [El +09]. Overviews of syntactic privacy model algorithms are provided by Fung et al. [Fun+10a] and Gkoulalas-Divanis, Loukides, and Sun [GLS14].

While both algorithms target k-anonymity, Machanavajjhala et al. [Mac+06] mention that it is simple to transform an algorithm for k-anonymity to l-diversity: Each check for k-anonymity, that is, checking if an equivalence class contains at least k data records, can be extended to l-diversity by additionally counting sensitive attributes in each equivalence class and checking the fulfillment of the condition for the chosen variant of l-diversity. Although not explicitly stated in the literature, a similar approach should be possible to achieve t-closeness by checking the distribution distance between the sensitive attributes in the equivalence class and in the whole dataset.

Incognito Algorithm

LeFevre, DeWitt, and Ramakrishnan [LDR05] introduce *Incognito*, an algorithm for achieving k-anonymity. It is based on *single-dimension full-domain generalization*, a variant of global recoding (see Section 2.5.1), meaning that attribute values are generalized independently from other attributes and in a way, that for each attribute all generalized

[15] Informally, an optimal solution looses as little information as possible when transforming a dataset so that it fulfills some syntactic privacy model. This can be formalized in different ways: Meyerson and Williams [MW04] use the smallest number of suppressed cells or attributes. Aggarwal et al. [Agg+05] employ the smallest number of suppressed cells or the cost of generalization in terms of the sum of the level of an attribute value in the generalization hierarchy divided by the total number of levels. LeFevre, DeWitt, and Ramakrishnan [LDR06] utilize the normalized average equivalence class size metric. Xiao, Yi, and Tao [XYT10] use the smallest number of suppressed cells or tuples with suppressed cells.

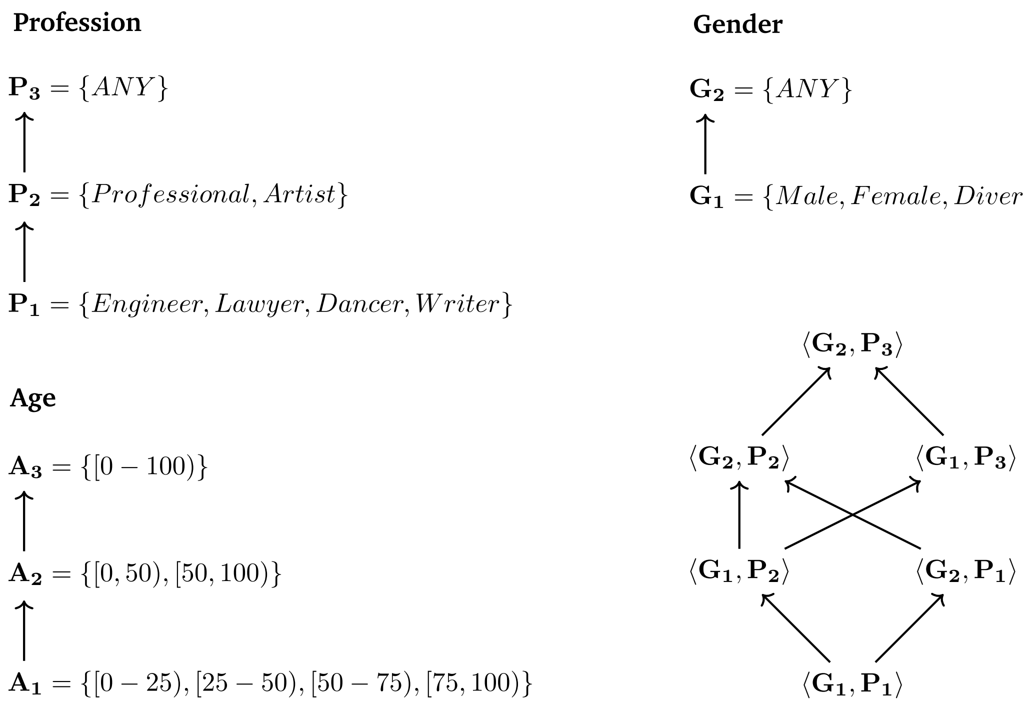

Profession
$\mathbf{P_3} = \{ANY\}$
$\mathbf{P_2} = \{Professional, Artist\}$
$\mathbf{P_1} = \{Engineer, Lawyer, Dancer, Writer\}$
Age
$\mathbf{A_3} = \{[0-100)\}$
$\mathbf{A_2} = \{[0,50),[50,100)\}$
$\mathbf{A_1} = \{[0-25),[25-50),[50-75),[75,100)\}$
Gender
$\mathbf{G_2} = \{ANY\}$
$\mathbf{G_1} = \{Male, Female, Diverse\}$
$\langle \mathbf{G_2}, \mathbf{P_3} \rangle$
$\langle \mathbf{G_2}, \mathbf{P_2} \rangle$
$\langle \mathbf{G_1}, \mathbf{P_3} \rangle$
$\langle \mathbf{G_1}, \mathbf{P_2} \rangle$
$\langle \mathbf{G_2}, \mathbf{P_1} \rangle$
$\langle \mathbf{G_1}, \mathbf{P_1} \rangle$

Figure 2.5: *Domain generalization hierarchies* for the attributes profession, age, and gender introduced in Figure 2.3 as well as a *multi-attribute generalization lattice* for the attributes profession and gender.

values are located in the same level of the generalization hierarchy. The authors further state the usage of optional tuple suppression, but provide no further details.

In the following we re-use the *value generalization hierarchies* introduced in Figure 2.3 for the attributes profession, age, and gender as an example. Since in full domain generalization the generalization is well defined by the level of the generalization hierarchy, we can simplify the hierarchy and just look at *domain generalization hierarchies*. Figure 2.5 shows the respective domain generalization hierarchies for the value generalization hierarchies depicted in Figure 2.3. When we describe the generalization of multiple attributes, this can be achieved by *multi-attribute generalization lattices*, as depicted in Figure 2.5 for the attributes profession and gender. This lattice describes possible generalization paths from the fully specialized (that is, not generalized) state to the point of full generalization.

Incognito uses an iterative process to find all possible k-anonymous full-domain generalizations. The algorithm starts with single QID attributes and iteratively checks larger subsets of all QID attributes for the fulfillment of k-anonymity with respect to given generalization lattices. In each iteration i the algorithm performs two operations:

- The iteration starts with a given graph of candidate multi-attribute generalizations with nodes C_i (containing QID sets of size i) and edges E_i. This graph can be seen as a set of generalization lattices, like the one depicted in Figure 2.5. The algorithm performs a breadth-first search for nodes which would result in a k-anonymous table. The resulting nodes are denoted S_i. This is done by starting at all root nodes of the graph and checking if the application of respective generalizations leads to a k-anonymous table. The authors use frequency sets for this purpose, which

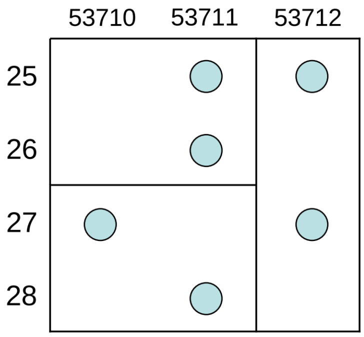


Figure 2.6: 2-anonymous multidimensional partitioning for QID attributes *age* and *zip code*. Image taken from [LDR06].

allows for faster computation of child nodes. If k-anonymity is fulfilled for the dataset at a node, than all child nodes also would lead to a k-anonymous dataset (*generalization property*), so they can directly be marked. The search is performed until all nodes are checked.

- Using S_i the algorithm constructs the graph (C_{i+1}, E_{i+1}). This construction is a three-step process. First, nodes from S_i, in other words, k-anonymous domain generalization values, are joined in a specific way to yield nodes representing domain generalization values for QID sets of size $i + 1$. Some of these joined nodes need to be pruned since they contain subsets of generalizations, which are not present in S_i, that is, which do not result in a k-anonymous table. Finally, the correct edges must be created with respect to the domain generalization hierarchies.

The first iteration uses all values of the QID attribute domain generalization hierarchies (examples are shown in Figure 2.5) as candidate nodes C_1 and their paths as edges E_1. The final iteration results in the set of all full-domain generalizations whose application results in a k-anonymous dataset. These candidates can be compared against each other regarding some notion of *minimality*, for example, based on the varying importance of some attributes being unaltered.

There exist variants of the Incognito algorithm which result in tables fulfilling l-diversity and t-closeness [Mac+06; LLV07].

Mondrian Algorithm

LeFevre, DeWitt, and Ramakrishnan [LDR06] present *Mondrian*, an algorithm for achieving k-anonymity via partitioning-based multidimensional recoding. In partition-based models, the attribute domains are partitioned and the data records are sorted into the respective partitions according to their attribute value. Therefore these models are especially well-suited for continuous (for example, numeric) data. An example for a k-anonymous multidimensional partition is shown in Figure 2.6.

Even though, as LeFevre, DeWitt, and Ramakrishnan show, optimal k-anonymous multidimensional partitioning as an NP-hard problem, with Mondrian they provide a

```
function MultiDimPartition(partition)
    if (no valid multidimensional cut for partition)
        return partition
    else
        dim ← choose_dimension()
        fs ← frequency_set(partition, dim)
        median ← find_median(fs)
        lPart ← {t ∈ partition: t.dim ≤ median}
        rPart ← {t ∈ partition: t.dim > median}
        return MultiDimPartition(lPart) + MultiDimPartition(rPart)
```

Listing 2.1: The multidimensional partitioning algorithm of Mondrian. Listing based on [LDR06].

heuristic algorithm and prove that it results in constant-factor approximations of the optimal solution. Mondrian proceeds in two steps:

1. The algorithm constructs multidimensional regions that cover the attribute domain space for all QID attributes by top-down greedy partitioning. For this purpose a recursive approach (depicted in Listing 2.1) is used. Starting with the whole attribute domain space, in each recursive call the space is partitioned in two parts with respect to a specific attribute and a split value as long as there is a partition which does not violate k-anonymity. The dimension and split value can be chosen in multiple ways. The authors propose to choose the dimension with the widest (normalized) value range and the median of attribute values for the chosen dimension as split value.

2. After partitioning, the algorithm computes summary statistics for the data records in each region, which represent the equivalence classes of the resulting dataset, and maps all data records in the region to these summary statistics.

There exists a variant of the Mondrian algorithm which results in a table fulfilling t-closeness [LLV10].

2.6.6 Further Privacy Models

While k-anonymity, l-diversity, and t-closeness are the most well-known privacy models, there is a variety of work constructing similar privacy models which tackle the same weaknesses or improve on these models in certain scenarios or with respect to certain attackers. Some examples are the following:

- k-Map [Swe01] is a relaxation of k-anonymity which requires k data records to share the same QID values – yet not with respect to to the published dataset but with respect to the whole population.

- δ-presence [NAC07] is a privacy model which protects the information whether an individual is part of a anonymized dataset.

- (α, k)-anonymity [Won+06] provides a similar approach to l-diversity in that it bounds the frequency of sensitive attributes in all equivalence classes of a k-anonymous table.

- (k, e)-anonymity [Zha+07] improves the privacy guarantees for k-anonymous data with respect to numerical sensitive attributes by requiring the range of the sensitive attribute values in each equivalence class to be at least e.
- m-confidentiality [Won+07] restricts the probability of successfully linking individuals and data records in a table as a countermeasure against a minimality attack (see Section 2.6.8).
- k^m-anonymity [TMK08] deals with transactional databases, in which multiple data records can belong to a single individual.
- (X, Y)-Privacy [WF06] generalizes k-anonymity for the scenario of multiple anonymized releases of the same dataset.
- m-Invariance [XT07] is a privacy model for the publication of multiple anonymized versions of a dynamic dataset.

Comprehensive overviews providing details to these examples and presenting further models can be found in several publications [Che+09; Fun+10a; De +12; GLS14; Zig+20; MH22].

2.6.7 Curse of High Dimensionality

The *curse of (high) dimensionality*[16] describes the problem that certain data mining tasks depending on the distance between elements, such as similarity search or nearest-neighbor clustering, are ineffective or inefficient in high-dimensional spaces due to the sparsity of elements in these spaces. Even the mere concepts of spatial locality, similarity, proximity, or distance are not useful under some assumptions in high-dimensional spaces. [Bey+99] has shown that the ratio of distances between nearest and farthest neighbors of elements approaches 1 in these situations. This problem is well-known in the data mining community [WSB98; IM98; Bey+99; AHK01].

The same limitation applies to k-anonymity and its successors, as Aggarwal [Agg05] has shown, since the concept of (utility-preserving) generalization depends on spatial locality. k-anonymity can be achieved by finding clusters of k similar data records and generalizing these data records to an equivalence class, so that the generalization does not unnecessarily impede the data utility. This intuition shows the direct relationship of k-anonymity and the nearest-neighbor problem and indicates the relevance of the curse of high dimensionality for syntactic privacy models. Aggarwal shows that in many high-dimensional scenarios, even for 2-anonymity, the loss of utility can render the results unacceptable for data mining tasks. Stadler, Oprisanu, and Troncoso [SOT22] formulate this insight in an inverse way: “information-rich datasets that are valuable for statistical analysis also always contain enough information to conduct privacy attacks.”

The curse of high dimensionality is also related to privacy breaches, in which the information in de-identified large sparse datasets is sufficient to re-identify individuals in these datasets. One example is the case of the *Netflix Prize dataset* presented in Section 2.8.4.

[16] Presumably, the term was introduced by Clarkson [Cla94].

Table 2.7: Vulnerability of covered syntactic privacy models to covered attacks.

Attack	k**-anonymity**	l**-diversity**	t**-closeness**
Homogeneity Attack	x		
Background Knowledge Attack	x		
Skewness Attack	x	x	
Similarity Attack	x	x	
Minimality Attack		x	x
Downcoding Attack	x	x	x
Attacks when publishing multiple datasets	x	x	x
- Unsorted Matching Attack			
- Complementary Release Attack			
- Temporal Attack			
- Composition Attacks			
- Correspondence Attacks			
deFinetti Attack	x	x	x

2.6.8 Attacks on Syntactic Privacy Models

In this section, we describe attacks on syntactic privacy models. Some of them are just applicable to specific models, while others pose general weaknesses in syntactic privacy models. Table 2.7 provides an overview of these attacks and their applicability to the privacy models covered before. A similar overview which includes more syntactic privacy models, but less detailed attack categories is provided by Zigomitros et al. [Zig+20].

Homogeneity Attack

The *homogeneity attack*, introduced by Machanavajjhala et al. [Mac+06], is possible when all records in an equivalence class share the same sensitive attribute. Even though an adversary might not be able to pinpoint the explicit record for an individual (identity disclosure), they can directly infer the value of the sensitive attribute for that individual (attribute disclosure).

An example for a vulnerable dataset is shown in Table 2.8. Even though the table provides k-anonymity for $k = 2$, one of the equivalence classes consists of HIV patients only. An adversary, who identified a target individual in this equivalence class, can directly deduce the sensitive value of that individual.

Table 2.8: A 2-anonymous dataset vulnerable to the homogeneity attack.

Sex	YOB	Postcode	Disease
f	1978–1979	54321	HIV
*	1975–1983	543**	Stroke
f	1978–1979	54321	HIV
*	1975–1983	543**	PAD
*	1975–1983	543**	Breast cancer

Table 2.9: A 2-anonymous dataset vulnerable to the background knowledge attack.

Sex	YOB	Postcode	Disease
f	1978–1979	54321	HIV
*	1975–1983	543**	Stroke
f	1978–1979	54321	PAD
*	1975–1983	543**	Ovarian cancer
*	1975–1983	543**	Breast cancer

Background Knowledge Attack

Machanavajjhala et al. [Mac+06] propose the concept of *background knowledge attacks*. Based on additional insights not given by the anonymized table, an adversary can deduce the sensitive value of an individual with a higher probability than promised by the privacy model. These additional insights (also referred to as *background knowledge*, *external knowledge*, *auxiliary information*, or *side information* [GKS08]) can comprise *instance-level background knowledge*, such as Alice showing symptoms of a specific disease, as well as *demographic background knowledge*, such as the prevalence of cancer in young individuals being low. An adversary with additional background knowledge might be able to violate the privacy guarantees of a specific model, when this knowledge is not considered during anonymization.

An example for a vulnerable table with respect to demographic background knowledge is given in Table 2.9. An adversary targeting a male patient with postcode 54328 can deduce that his target suffers from a stroke with high probability, since men usually do not suffer from breast or ovarian cancer.

The main problem when dealing with this type of attack is that it is often hard or even impossible to accurately model the adversary's background knowledge (or more precisely, the background knowledge of each and every potential adversary). Furthermore, the background knowledge of an adversary can change over time, for example, when further datasets are published or research publishes new findings about correlations between QIDs and sensitive attributes.

Tao et al. [Tao+08] present a specific type of background knowledge attacks, called *corruption attacks*. These attacks deal with a special type of background knowledge: A *corruption* describes the situation in which an adversary has learned the sensitive attribute of an individual I from a different source than the published table T. This knowledge can put other individuals in the same equivalence class as I in danger since knowledge about corruptions can potentially render the guarantees of the privacy model meaningless.

Martin et al. [Mar+07] propose a theoretical framework to capture all pieces of background knowledge an adversary might possess in form of a formal language. Based on this language, they provide algorithms for the efficient calculation of worst-case disclosure risks for adversaries bounded in the number of background information pieces. Even though Martin et al. show theoretically interesting results, it is unclear how to these results would translate into practice.

Skewness Attack

Li, Li, and Venkatasubramanian [LLV07] present the *skewness attack* applicable to situations in which the global distribution of sensitive attributes is skewed. If an equivalence class contains a different distribution of sensitive values in comparison to the overall distribution, this presents a privacy risk for individuals in this class.

An example given by Li, Li, and Venkatasubramanian [LLV07] deals with the result table for a medical test in which just a small minority of individuals tests positive for a particular virus. For the sake of convenience we assume a prevalence of 0.01 in the test population. An equivalence class of size $n = 50$ with 49 positive and 1 negative cases would fulfill distinct 2-diversity[17], but would pose a serious threat to an individual for which the probability of a positive test is now 0.98.

Similarity Attack

The similarity attack, introduced by Li, Li, and Venkatasubramanian [LLV07], is possible when multiple sensitive attribute values are semantically similar. Even though the specific sensitive attribute value remains protected, an adversary could deduce the "generalized" type of sensitive attribute when all records in an equivalence class share semantically similar values.

An example is shown in Table 2.10. All patients in one equivalence class suffer from a stomach-related disease. Even though an adversary does not learn the detailed disease, they still find out about the specific class of diseases for the target.

A similar attack, called *proximity attack*, is described by Li, Tao, and Xiao [LTX08]. It is possible when an adversary can deduce the narrow interval, in which the numerical sensitive attribute of a victim falls, with high confidence.

[17] Similar considerations for entropy diversity and (c, l)-diversity are provided by Machanavajjhala et al. [Mac+06] as well.

Table 2.10: A dataset vulnerable to the similarity attack.

Sex	YOB	Postcode	Disease
f	1978–1979	54321	Gastritis
*	1975–1983	543**	Flu
f	1978–1979	54321	Gastric ulcer
*	1975–1983	543**	PAD
*	1975–1983	543**	Breast cancer

Table 2.11: A dataset vulnerable to the minimality attack.

Name	Age
Alice	48
Bob	44
Carol	47
Dave	51
Eve	58
Frank	53
Grace	54

(a) The adversaries background knowledge.

Age	Disease
40–60	HIV
40–60	HIV
40–60	Flu
40–60	Skin cancer
50–60	PAD
50–60	Breast cancer
50–60	HIV

(b) The distinct 3-diverse table.

Minimality Attack

Wong et al. [Won+07] introduce the concept of *minimality attacks*. This class of attacks are based on the fact that most algorithms for syntactic anonymity rely on the so-called *minimality principle*: The algorithm should not alter (for example, generalize) the data more than necessary to achieve the desired privacy model and therefore minimize the utility loss. Minimality attacks exploit this principle. Given that an adversary has knowledge about the used algorithm, they can draw conclusions about algorithm steps which lead to the anonymized dataset and with that also about the original dataset. The minimality attack then allows to increase the adversary's belief about the sensitive value of an individual.

An example is shown in Table 2.11. We assume an adversary with full background knowledge as given by Table 2.11a. When the *distinct 3-diverse* (see Section 2.6.3) dataset depicted in Table 2.11b is published, the adversary can deduce that the records of age 40-50 did not fulfill 3-diversity and had to be generalized again. Therefore, both records having HIV as the sensitive attribute value must be part of this age group. This increases the adversary's belief about Alice suffering from HIV to $\frac{2}{3}$ (assuming no further background knowledge) which violates the guarantees from l-diversity. This inference would have not been possible without an anonymization algorithm following the minimality principle.

Table 2.12: A 2-anonymous dataset vulnerable to the downcoding attack.

Sex	Postcode	Disease
$*_1$	54321	Flu
$*_2$	54321	Stroke
$*_3$	54328	HIV
$*_4$	54328	PAD

Wong et al. provide examples on how to attack a variety of privacy models including l-diversity and t-closeness. They introduce the concept of *m-confidentiality* and an algorithm for it to protect against minimality attacks.

As Cormode et al. [Cor+10] state, this attack makes strong assumptions about an adversary's knowledge: They are assumed to know the QID values for all data records, the anonymization method, and the anonymized data. Cormode et al. find three properties making algorithms vulnerable to minimality attacks: deterministic behavior, asymmetric equivalence class choices, and considering QIDs and sensitive attributes together during anonymization. Indeterminism, symmetric equivalence class choices, or focusing on QIDs or sensitive attributes only (as all k-anonymity algorithms proceed) can reduce or completely prevent the vulnerability of algorithms to the minimality attack. Furthermore, the authors examine an example algorithm (*Greedy Grouping*) which should be strongly vulnerable to minimality attacks and show that the increase in an adversary's posterior belief (and therefore the criticality of the attack) is bounded by $\frac{e}{l}$ with $e \approx 2.718828$ being Euler's number and l being the syntactic privacy parameter of l-diversity. They conclude that the effect of the minimality attack for larger datasets and larger privacy parameters is less dramatic than in the small examples given by Wong et al.

Downcoding Attack

Cohen [Coh22] introduces the concept of *downcoding attacks* which also take advantage of the minimality principle. The downcoding attack allows to undo anonymization by minimal hierarchical generalization and to recover some fractions of the generalized data. In comparison to the minimality attack, the downcoding attack does work against k-anonymity as well.

The foundation of the downcoding attack is the observation that minimality leaks information. A simple example is given in Table 2.12. Since the k-anonymization with $k = 2$ in this example is assumed to be minimal, an adversary can infer that each generalization step was necessary in the sense that the equivalence class before the generalization did violate k-anonymity. Therefore, without any background knowledge, the adversary learns that $\{*_1, *_2\} = \{*_2, *_3\} = \{m, f\}$. An equivalence class with two female or male patients is not possible due to minimality. Given a data distribution from which the data records are drawn, Cohen proofs that inferences of this kind can be exploited to recover generalized data values with non-negligible probability.

Cohen provides an example on an artificial dataset of clustered Gaussian distributed data. But several questions remain open for future work: It is unclear, in which settings regarding data distributions and generalization hierarchies downcoding attacks are possible or provably impossible and if we can assess the vulnerability against downcoding attacks. Finally, the applicability of downcoding attacks on real-world datasets has not been demonstrated at the moment.

Attacking Multiple Published Datasets

The covered privacy models are concerned with singular publications of anonymized datasets. In comparison to this scenario of *static anonymization*, *dynamic anonymization* deals with anonymizing a potentially updated dataset multiple times [HBN11]. In this scenario additional challenges and weaknesses arise.

In one of the early publications about k-anonymity Sweeney already provides some of these weaknesses [Swe02]. Let T denote the original dataset and T_1^*, T_2^* two different anonymized versions of T. The *Unsorted Matching Attack* is possible, when the order of data records in T_1^* and T_2^* remains the same as in the original dataset. By linking all records $t_1^{(i)} \in T_1^*$ and $t_2^{(i)} \in T_2^*$ via their index i an adversary can obtain more information about an individual than the individual anonymized tables allow. The *Complementary Release Attack* describes an attack where the records of T_1^* and T_2^* contain some attributes which are not considered to be part of the QIDs. Then an adversary might be able to link records in these two anonymized datasets via these attributes and to single out unique individuals, violating the guarantees of k-anonymity. The *Temporal Attack* is enabled when T_1^* and T_2^* are anonymized versions of T at different moments and T is dynamically changing between these moments. Since the later anonymization does not respect the earlier one, linking the tables can reveal sensitive information about new records as well as about already present records. This is caused by the potentially different anonymization due to the influence of new records.

Ganta, Kasiviswanathan, and Smith [GKS08] introduce the concept of *composition attacks* – attacks which are possible when individuals appear in multiple anonymized datasets. They provide the *intersection attack*, a simple example of a composition attack, which works when the anonymization methods preserve sensitive values (*exact sensitive value disclosure*) and allow to find the equivalence class of an individual based on their QIDs (*locatability*). An adversary can compute the intersection of sensitive values for an individual over all datasets. This potentially allows to increase the adversary's belief for sensitive attribute values in comparison to the individual datasets or even to single out a specific value. Composition attacks can be seen as background knowledge attacks with independently anonymized datasets as background knowledge.

Fung et al. [Fun+08] provide the concept of *correspondence attacks*. These are possible in a scenario in which (potentially differently) anonymized versions of a growing dataset are released at multiple points in time. It is assumed, that records are not deleted and all releases contain all existing records present at that moment in time. Records which exist at times T_1 and T_2 are therefore present in respective data releases R_1 and R_2 and a record $r_1 \in R_1$ always has a counterpart $r_2 \in R_2$ called the *corresponding record*. This fact can be exploited by an adversary in possession of both data releases in different ways. Fung et al. propose three types of correspondence attacks (*Forward-attack*, *Cross-attack*, *Backward-attack*) which depend on the time the target individual was included in the dataset and the release to attack. Furthermore, they provide the notion of *BCF-anonymity* and an algorithm for it to deal with correspondence attacks.

Byun et al. [Byu+06] look at the same problem (while calling the type of attacks *inference attacks*). In comparison, their solution requires to withhold some records in later releases under special circumstances. A similar problem is studied by Xiao and Tao [XT07]. In comparison they also include the possibility to delete records between data releases. They provide the principle of *m-invariance* to protect subsequent releases. He, Barman, and Naughton [HBN11] extend these attack variants by the *equivalence attack* which additionally covers inferences drawn with respect to the equivalence of sensitive attribute sets for sets of individuals.

The deFinetti Attack

The *deFinetti attack* (named after De Finetti's theorem from probability theory), invented by Kifer [Kif09], takes advantage of the fact that syntactic anonymization methods preserve statistical correlations between QIDs and the sensitive attributes. A classifier trained on the anonymized data can be used to predict the sensitive attributes of an individual in the dataset with a much higher certainty than assumed by privacy models. These models generally base their privacy guarantees on simplified assumptions. For example, in the *random worlds model* an adversary computes the probability of sensitive attribute values for an individual by looking at the unweighted frequency of values in all possible worlds. Kifer shows that a well-trained classifier allows for way better predictions of a sensitive attribute value than expected with these simplified assumptions on the example of a Naive Bayes classifier and the *Anatomy* anonymization method (see Section 2.5.5).

Cormode [Cor11] shows a similar attack for DP (see Section 2.7) by using a Naive Bayes classifier computed on differentially private histograms for all QIDs and the sensitive attributes. Furthermore, Cormode states that the deFinetti attack loses its accuracy for moderate parameters of l-diversity and that certain parameter choices can make DP even more suspectible to this attack. While Kifer blames the immaturity of privacy models for the success of the attack, Cormode makes a more fundamental argument: The release of anonymized data can reveal statistical correlations of sensitive attributes in the population – and often this is even the objective of data collection and publication[18]. Therefore this attack falls under the category of *probabilistic disclosure* (see Section 2.1) and Cormode argues for making decisions with respect to a tolerable level of potential adversary's belief change.

2.6.9 Concluding Remarks

In comparison to the application of simple de-identification techniques (see Section 2.5), syntactic privacy models, like k-anonymity or t-closeness, introduce measurable guarantees regarding specific threats. They reduce the disclosure risks in comparison to the

[18] Cormode provides the example of scientific studies being presented in the form of conditional probabilities, such as *Drinking two glasses of wine each day reduces the chance of heart disease by 50 %.*

original data and provide *truthfulness* on the data record level in the sense that the original attribute values are kept or are replaced by generalizations of these values [CT13]. But these models also bring major problems with them:

- One of the central issues is the need to classify the dataset attributes into the categories identifying attribute, QID, sensitive attribute, or nonsensitive attribute, as covered in Section 2.6.1. The assessment of attributes must incorporate potential background knowledge of possible adversaries and it is by no means obvious how to do this [De +12]. Missing attributes when determining QIDs can lead to high re-identification risks.

- On the other hand, classifying all attributes as QIDs – especially for higher-dimensional data – can severely impact the utility of a dataset due to the curse of high dimensionality (see Section 2.6.7).

- Furthermore, there is a variety of attacks against syntactic privacy models (see Section 2.6.8). While some of these are specific to a particular model, others are generally applicable to most models. Additionally, later models fix weaknesses of earlier models often at the expense of higher utility losses. For example, t-closeness is not suspectible to some attacks against k-anonymity and l-diversity but comes with higher utility costs (see Section 2.6.4).

- While the privacy parameters, such as k in k-anonymity, especially in comparison to ε in DP, carry some intuitive meaning with respect to potential disclosure risks, it is often unclear how to choose suitable values for actual scenarios.

- Finally, as shown in Section 2.6.8, these models generally do not allow for composition.

2.7 Differential Privacy

With the observed weaknesses and subtleties in the application of syntactic privacy models, there was the need for a paradigm shift. This happened with the upcoming of semantic privacy models and their most well-known representative DP, which was introduced by Dwork [Dwo+06b; Dwo06]. While there are further related semantic privacy models (see Section 2.7.6), the concepts are quite similar. Therefore we focus on DP as the most well-known representative.

The basic idea of DP is to reduce the impact, a single individual has on the result of a computation, by introducing a specific amount of randomness into the computation. This indicates a fundamental difference to syntactic anonymity models: DP is not a property of a published dataset (or, more generally, the result of a computation) but of the computation itself.

Figure 2.7 provides a visual intuition of DP. We look at two databases D_1 and D_2, which differ in a single data record. DP guarantees, that a computation performed by a (probabilistic) differentially private mechanism M on both databases produces similar results. This minimizes the influence a single individual has on the result of a

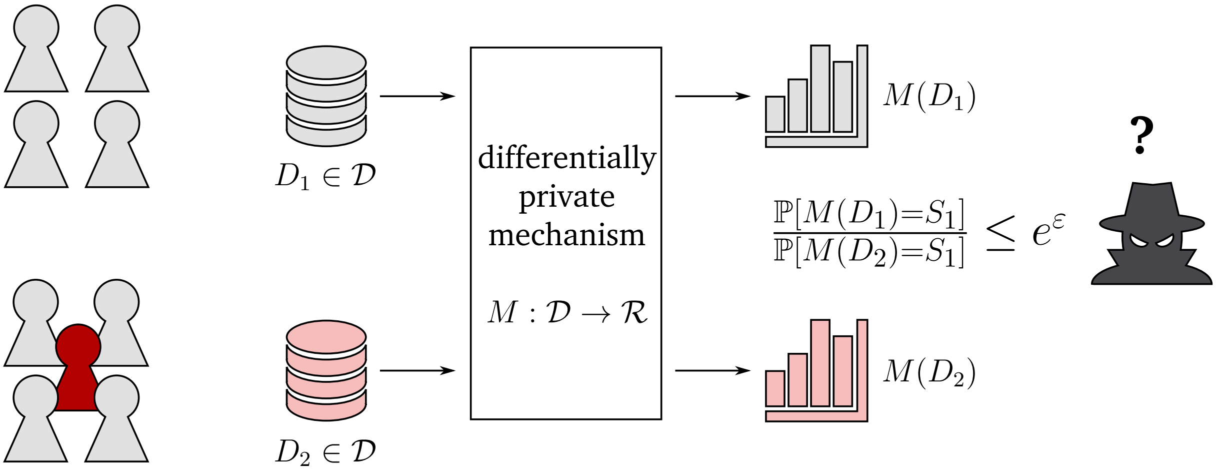

$D_1 \in \mathcal{D}$
$D_2 \in \mathcal{D}$
differentially private mechanism
$M : \mathcal{D} \to \mathcal{R}$
$M(D_1)$
$M(D_2)$
$\frac{\mathbb{P}[M(D_1)=S_1]}{\mathbb{P}[M(D_2)=S_1]} \leq e^{\varepsilon}$
?

Figure 2.7: Intuition for ε-DP. Computing the result of a (probabilistic) mechanism M on two databases D_1, D_2 which differ in a single individual (depicted in red) should provide an adversary with minimal information about the influence of the individual's data.

computation and prevents adversaries from drawing conclusions about this individual by looking at the computation result[19].

DP provides several properties which syntactic privacy models generally lack [Hsu+14; DKM19]: It enables us to quantify the influence a single individual has on the result and respectively the risk the individual is exposed to by publishing the result of the computation. The privacy guarantees do not depend on adversary's background information, capabilities, or goals – and this holds even for future adversaries. And, as we will see in Section 2.7.2, multiple computations are composable and the resulting risk is measurable as well.

In this section we provide a detailed introduction to DP. Section 2.7.1 will formalize the given intuition and lead to the notion of ε-DP. In Section 2.7.2 we look at composability properties ε-DP offers. Section 2.7.3 introduces differentially private mechanisms which translate the concept to specific computations. The following Section 2.7.4 looks at another often used variant of DP, called (ε, δ)-DP. In practice, it is relevant how to choose the privacy parameters ε and δ. Several approaches for this purpose are surveyed in Section 2.7.5. Because we just cover the most relevant DP basics in this section, in the final Section 2.7.6 we provide an overview of further DP-related topics as a starting point for research in the field of DP.

[19] To be more precise, the results of a computation let an adversary infer nothing about the individual that could not be inferred without the individual participating in the computation. So DP does not protect against, such as population disclosure (see Section 2.1).

2.7.1 ε-Differential Privacy

In this section we formalize the given intuition for DP. The following definitions originate from [DR+14]. We begin with defining databases and the distance between databases, for which a histogram-based definition is used. A *data record* is represented as an element x of the base set of all possible elements $\mathcal{X}$[20]. A *database* can be interpreted as a histogram $\mathbb{N}^{|\mathcal{X}|}$ which contains the count of respective data records (elements from the base set). For readability reasons, we refer to this domain as $\mathcal{D}$.

This histogram-based definition allows us to define the distance between two databases with respect to the l_1-norm $\|\cdot\|_1$ (the so-called *manhattan distance*). The l_1-norm of a single database D is defined as

$$\|D\|_1 = \sum_{i=1}^{|\mathcal{X}|} |D^{(i)}|,$$

which can be understood as the number of elements in this database. The *distance between two databases* D_1, D_2 is then naturally given by $\|D_1 - D_2\|_1$. This results in a histogram of count differences between D_1 and D_2 and describes in how many data records the databases D_1 and D_2 differ. With these preliminary definitions, we can formalize ε-DP.

Definition 2.7.1. A randomized algorithm M with domain $\mathcal{D}$ is called *ϵ-differentially private*, if for all S in the range $\mathcal{R}$ of M and for all $D_1, D_2 \in \mathcal{D}$ with $||D_1 - D_2||_1 \leq 1$

$$Pr[M(D_1) \in S] \leq e^{\varepsilon} Pr[M(D_2) \in S]$$

holds. The probability space follows from the randomization of M.

Given this definition, one might ask how it relates to the informal intuition presented in the beginning of this section. The database distance describes two databases which differ in a single individual's data, which can be understood as looking at the same database with and without the data of an individual[21]. For a differentially private mechanism M and a potential result S the probability for S being the result of M computed on these two databases must not differ by more than e^{ε}. This captures our informally described intuition of M *producing similar results on both databases*. Figure 2.7 shows this relation for a single result $S \in \mathcal{R}$. The parameter ε presents a way to mediate between privacy and data utility. A smaller value leads to smaller deviations of algorithm results with respect to databases D_1 and D_2, that is it reduces the influence of individual data records on the result and therefore increases the privacy for individuals. Section 2.7.5 presents approaches for choosing an adequate parameter value for ε. The next section covers convenient properties which directly follow from defining privacy in this way.

[20] We observe the theoretical nature of this definition. In practice it would be quite laborious to list all possible database elements. However, this definition allows for a clean understanding of database distances.

[21] Formally, one can differentiate between *bounded* and *unbounded DP* here [KM11]. Bounded DP describes a definition in which a single tuple in the database is changed, while in unbounded DP D_2 is obtained from D_1 by adding or removing a tuple.

2.7.2 Properties of Differential Privacy

An important reason for the relevance of DP is the possibility to quantify the privacy loss even under composition of multiple queries (in contrast to syntactic mechanisms). The following sections cover properties of DP under composition and with respect to postprocessing.

Sequential Composition

If several queries are executed on the same database, the achieved level of privacy is inevitably reduced. For example, multiple executions of the Laplace mechanism for the same type of query would potentially allow to deduce the real value with a high probability. An advantage of DP is the possibility to quantify by how much the privacy guarantee is weakened. For two queries $A(D)$ and $B(D)$ with privacy parameters ε_A and ε_B the combined query $C(D) = (A(D), B(D))$ fulfills ε_C-DP with $\varepsilon_C = \varepsilon_1 + \varepsilon_2$. This property is called *sequential composition* [McS09].

The proof for the sequential composition property is simple. The ratio of probabilities for a result (r_A, r_B) of the combined query is

$$\begin{aligned}
&\frac{\mathbb{P}[C(D_1) = (r_A, r_B)]}{\mathbb{P}[C(D_2) = (r_A, r_B)]}\\
=&\frac{\mathbb{P}[A(D_1) = r_A]\mathbb{P}[B(D_1) = r_B]}{\mathbb{P}[A(D_2) = r_A]\mathbb{P}[B(D_2) = r_B]}\\
=&\left(\frac{\mathbb{P}[A(D_1) = r_A]}{\mathbb{P}[A(D_2) = r_A]}\right) \cdot \left(\frac{\mathbb{P}[B(D_1) = r_B]}{\mathbb{P}[B(D_2) = r_B]}\right)\\
\leq&\exp(\varepsilon_A)\exp(\varepsilon_B)\\
=&\exp(\varepsilon_A + \varepsilon_B).
\end{aligned}$$

Because of the sequential composition property the privacy parameter ε often is referred to as *privacy (loss) budget*. The privacy budget for n queries, which in combination guarantee ε-DP, can be split over these n queries by using n privacy parameters $\varepsilon_1, \ldots, \varepsilon_n$ with $\sum_{i=1}^{n} \varepsilon_i = \varepsilon$ for the queries. Further results for the influence of composition on the achievable privacy level are provided by Dwork, Roth, et al. [DR+14], Murtagh and Vadhan [MV16], and Vadhan and Wang [VW21]. These results allow a better choice of privacy parameters when using multiple queries and extend the given property to (ε, δ)-DP.

Parallel Composition

Another composition property of DP is called *parallel composition* [McS09]. This property is based on the idea that, when performing queries on disjoint subsets of the database, a single individual can just contribute to one of these queries. Let $D^1, \ldots, D^n$ be disjoint subsets of the database D. For queries $M_1(D^1), \ldots, M_n(D^n)$, where each query $M_i(\cdot)$

guarantees ε-DP, the combination of all queries also guarantees ε-DP. For queries with differing privacy parameters the combination fulfills ε_{max}-DP with ε_{max} being the maximum privacy parameter over all queries. A proof for this property is provided by McSherry [McS09].

In comparison to sequential composition, which would give us a guarantee of $n\varepsilon$-DP, this allows for a much better estimation of the achieved privacy for the scenario of disjoint subsets. A natural application example for parallel composition are histogram queries (see Section 2.7.3).

Postprocessing

Often, there is a necessity to postprocess the result of a differentially private query before outputting it. For example, it can be required to round values or to substitute 0 for negative values in a histogram query. The following result shows that such subsequent, data-independent transformations of a DP mechanism do not violate the DP guarantees.

Given a deterministic postprocessing function $g : R \to R'$ and a DP mechanism $M : \mathcal{D} \to R$, then $g \circ M : \mathcal{D} \to R'$ is also differentially private. The following proof is provided by Dwork, Roth, et al. [DR+14]. Given two databases D_1, D_2 with $\|D_1 - D_2\|_1 \leq 1$ and a result $S \subseteq R'$ of the mechanism M as well as $T = \{r \in R : g(r) \in S\}$, we obtain

$$\begin{aligned}
\mathbb{P}[g(M(D_1)) \in S] &= \mathbb{P}[M(D_1) \in T] \\
&\leq \exp(\varepsilon)\mathbb{P}[M(D_2) \in T] \\
&= \exp(\varepsilon)\mathbb{P}[g(M(D_2)) \in S].
\end{aligned}$$

This result shows that each data-independent transformation of a differentially private mechanism result does not violate the guarantees of DP – this property holds completely independent of any additional information an adversary may possess.

2.7.3 Differentially Private Mechanisms

After presenting the principle of DP, this section will introduce examples of mechanisms that fulfill ε-DP. The main approach used in these methods is to add random noise to the result of a calculation. The amount of added noise, which, for example, can be obtained from a Laplace distribution, depends on the privacy parameter ε.

Randomized Response

A simple mechanism which fulfills DP is the *randomized response* mechanism. It was introduced for study questions, for which respondents are not willing to give correct answers, or answers at all, due to the sensitivity of the question topic. The basic idea is to introduce randomness in the response process to encourage respondents to answer honestly by providing some sort of *plausible deniability*. The concept has been used in the social sciences way before DP was introduced [War65; Gre+69]. Later it has been shown that it is a differentially private mechanism.

As a simple example, we look at a question with binary answers, for example: *Have you been smoking marijuana in the last week?*. For answering the question, respondents are asked to covertly flip a coin. For *heads* they should provide the real answer to the question. For *tails* they flip the coin again and answer with *yes* for *heads* and *no* for *tails*. In this way the interviewer cannot deduce the respondents real answer with certainty. But a large number of answers still allows statistical conclusions about the response distribution in the study population.

It is easy to show that this randomized response mechanism fulfills DP [DR+14]. Looking at a single answer for a respondent who would answer the original question with *yes*, they would answer *yes* when the first coin shows *heads* ($\mathbb{P} = \frac{1}{2}$) as well as when the first coin flip shows *tails* and the second one shows *heads* ($\mathbb{P} = \frac{1}{4}$). Therefore we obtain the conditional probability $\mathbb{P}[result = yes \mid truth = yes] = \frac{3}{4}$. Similar considerations can be made about the other relevant conditional probabilities. When we compare these probabilities, we obtain

$$\frac{\mathbb{P}[result = yes \mid truth = yes]}{\mathbb{P}[result = yes \mid truth = no]} = \frac{\mathbb{P}[result = no \mid truth = no]}{\mathbb{P}[result = no \mid truth = yes]} = \frac{\frac{3}{4}}{\frac{1}{4}} = 3.$$

Comparing this to the definition of ε-DP, we can directly see that this particular mechanism is $\ln 3$-differentially private. Other values of ε, and therefore more or less privacy for respondents, can be obtained by using an unfair coin, where heads and tails probabilities $\mathbb{P} \neq 0.5$.

This mechanism is not just a toy example for a DP mechanism, but is used as a base for large-scale DP implementations, for example, in the *RAPPOR* framework by Google [EPK14]. Further, more theoretical results related to randomized response and DP are provided by Kasiviswanathan et al. [Kas+11].

Laplace Mechanism

A quite natural way to achieve a differentially private version of a function $f : \mathcal{D} \to \mathbb{R}^k$ arises from the Laplace distribution in the form of the *Laplace mechanism* [Dwo+06b]. For this we simply compute

$$\mathcal{M}_{\text{Laplace}}(D, f(\cdot), \varepsilon) = f(D) + (Y_1, \ldots, Y_k),$$

in which Y_i represents a random variable drawn from the Laplace distribution $Lap(\frac{\Delta f}{\varepsilon})$. Δf denotes the so-called *sensitivity* of function f – a value that describes the maximum change a single individual can cause in the output of f[22]:

[22] Given here is the so-called global sensitivity, that is, the maximum change an individual can cause in two of all possible databases. This kind of sensitivity can potentially lead to the addition of much noise. There are other definitions of sensitivity, such as *smooth sensitivity* [NRS07], which relax the definition of global sensitivity. Further references to other sensitivity definitions are presented by Desfontaines and Pejó [DP20].

$$\Delta f = \max_{\substack{D_1, D_2 \in \mathcal{D} \\ \|D_1 - D_2\|_1 = 1}} \|f(D_1) - f(D_2)\|_1 .$$

Again, $\|\cdot\|_1$ describes the l_1-norm of the function results[23]. The used scale of the Laplace distribution $\frac{\Delta f}{\varepsilon}$ is responsible for its spread. For functions with a higher sensitivity Δf or stronger privacy guarantees, in other words, smaller values for ε, the probability of adding more noise is higher.

The following proof is presented by Dwork, Roth, et al. [DR+14]. We can show that the Laplace mechanism fulfills ε-DP by comparing the probability density functions for two databases $D_1, D_2 \in \mathcal{D}$ with $\|D_1 - D_2\|_1 \leq 1$. The probability density function of the Laplace distribution is formed by $p(x) = \frac{1}{2\sigma} e^{\frac{|x|}{\sigma}}$. We compare the probability densities p_{D_1} for algorithm $\mathcal{M}_{\text{Laplace}}(D_1, f(\cdot), \varepsilon)$ and p_{D_2} for algorithm $\mathcal{M}_{\text{Laplace}}(D_2, f(\cdot), \varepsilon)$ at an arbitrary point $z \in \mathbb{R}^k$:

$$\begin{aligned}
\frac{p_{D_1}(z)}{p_{D_2}(z)} &= \prod_{i=1}^{k} \left(\frac{\exp\left(-\frac{\varepsilon \cdot |f(D_1)_i - z_i|}{\Delta f}\right)}{\exp\left(-\frac{\varepsilon \cdot |f(D_2)_i - z_i|}{\Delta f}\right)} \right) \\
&= \prod_{i=1}^{k} \exp\left(\frac{\varepsilon \cdot (|f(D_2)_i - z_i| - |f(D_1)_i - z_i|)}{\Delta f} \right) \\
&\overset{(1)}{\leq} \prod_{i=1}^{k} \exp\left(\frac{\varepsilon \cdot |f(D_1)_i - f(D_2)_i|}{\Delta f} \right) \\
&= \exp\left(\frac{\varepsilon \cdot \sum_{i=1}^{k} |f(D_1)_i - f(D_2)_i|}{\Delta f} \right) \\
&= \exp\left(\frac{\varepsilon \cdot ||f(D_1) - f(D_2)||_1}{\Delta f} \right) \\
&\overset{(2)}{\leq} \exp(\varepsilon).
\end{aligned}$$

Inequality (1) follows from the triangle inequality[24]. Inequality (2) uses the fact $||D_1 - D_2||_1 \leq 1$ and the definition of the sensitivity

$$\frac{||f(D_1) - f(D_2)||_1}{\Delta f} \leq 1.$$

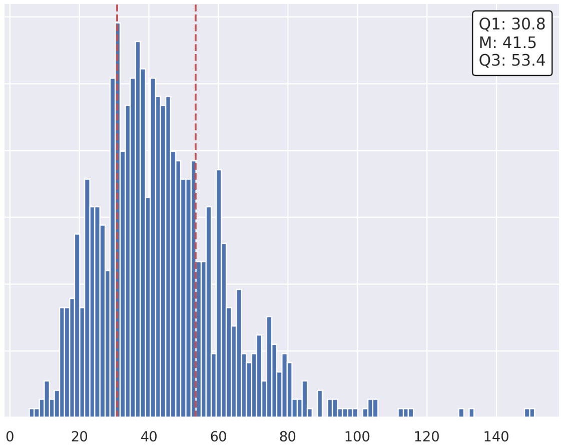
[23] A distance metric is used here, since the function results can potentially be multidimensional. This is especially relevant in comparison to the Gaussian mechanism introduced in Section 2.7.4 in which a different distance metric is used. However, for the following examples in this section this plays no role since they all are based on functions with single-dimensional results.

[24] More precisely, the inequality follows from a variant of the triangle inequality, we achieve by some transformations:

$$|A + B| \leq |A| + |B| \iff |A - C| - |B - C| \leq |B - A|.$$

Name	**Income** (k€)
Alice	40
Bob	17
Carol	123
Dave	51
Eve	59
...	...

(a) The income microdata.



(b) The distribution of incomes. This figure (and the following ones) additionally show the medians as well as the first and third quartiles for the distribution.

Figure 2.8: The artificial income dataset.

This general formulation of the Laplace mechanism can be used to obtain differentially private variants of different functions.

In the following sections we use an artificial dataset of income microdata D_{Income} to show the influence of DP to specific queries. For this dataset 1,000 data records with random[25] income values were created. An excerpt from this dataset and the distribution of income values of the whole dataset are shown in Figure 2.8a and Figure 2.8b. In the following mechanism descriptions based on the example we omit the digits below the thousands to reduce visual clutter. For the upcoming example queries we use a fixed privacy parameter $\varepsilon = 0.5$ (see Section 2.7.5 for how to choose reasonable privacy parameters).

Counting queries A *counting query* computes the number of individuals $D^{(i)}$ in a database D which fulfill a certain property (described by the indicator variable $\mathbb{I}(\cdot)$):

$$f_{count}(D) = \sum_{i=1}^{|D|} \mathbb{I}(D^{(i)}).$$

For example, this could be the number of people suffering from a specific type of cancer. To construct a differentially private counting query, we can resort to the Laplace mechanism. A single individual can change the result of a counting query by 1 at most, therefore the sensitivity of a counting query $\Delta f = 1$. Therefore it is sufficient to add random noise drawn from $Lap(\frac{1}{\varepsilon})$ to the original result of the counting query.

When considering our income dataset an example query is counting the number of individuals with an income above 60.000 €. The real result of the query $f_{count}(D_{Income}) =$

[25] The income values were drawn from the Gumbel distribution, which can be used to model the chances of natural disasters. Similarities are completely coincidental – but undeniable. It may be worthwhile for future economic studies to explore the question whether the current income distributions represent a special kind of natural disaster.

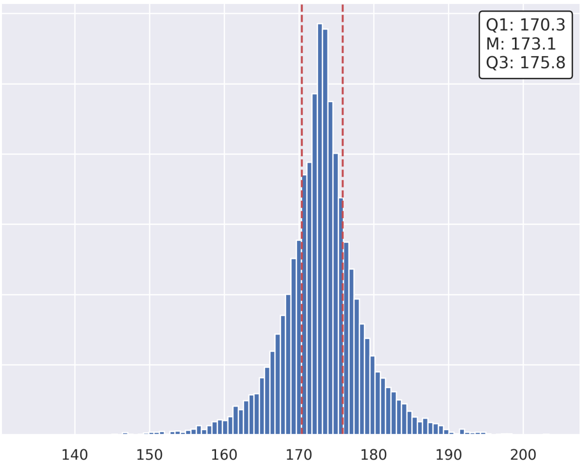


Figure 2.9: The distribution of differentially private counting query results for individuals with an income above 60,000 €.

173. Adding noise from $Lap(\frac{1}{\varepsilon})$ with $\varepsilon = 0.5$ does alter this value slightly. Figure 2.9 shows the distribution of differentially private results of this query. As we can see, most of the results fall into the range of 170 to 176. So the query should provide an result accurate enough for most use cases.

Histogram queries A *histogram query* divides the database D based on certain criteria into distinct subsets H_i and provides the number of individuals in these subsets. An example would be the age distribution of individuals in a database in ranges of 5 years:

$$f_{hist}(D) = (H_1, \ldots, H_n) \text{ with } H_i = \sum_{j=1}^{|D|} \mathbb{I}(D^{(j)}, H_i).$$

The indicator variable $\mathbb{I}(D^{(j)}, H_i)$ describes the (non-)membership of an individual $D^{(j)}$ in the specific subset H_i. A single individual belongs to exactly one of the subsets and does influence the count for this subset by 1 – an example for the parallel composition property of DP (see Section 2.7.2). Therefore, just as in the case of simple counting queries, the sensitivity of the query $\Delta f = 1$. Similarily, we can use the Laplace mechanism by adding random noise drawn from $Lap(\frac{1}{\varepsilon})$ to each subset count.

Table 2.13a shows the histogram $f_{hist}(D_{Income})$ for our income database and Table 2.13b a possible differentially private result of the query is shown. Noise from the Laplace distribution $Lap(\frac{1}{\varepsilon})$ with $\varepsilon = 0.5$ was added independently to each of the bins. Additionally all resulting counts were rounded to the next integer. This does no harm to the DP guarantees due to the postprocessing property (see Section 2.7.2). Notice, that the result in Table 2.13b is just one example and does not show the result distribution. However, since the amount of noise is the same as in the counting query covered before and the noise is drawn independently for each income range, the distribution for each differentially private count value follows the one covered in Figure 2.9 (obviously with respective median values).

Table 2.13: An example for a differentially private histogram query.

Income	Count
0–30	221
30–60	606
60–90	153
90–120	16
120–150	3

(a) The unaltered income range histogram query result.

Income	Count
0–30	224
30–60	606
60–90	151
90–120	20
120–150	4

(b) Example result for the differentially private histogram query.

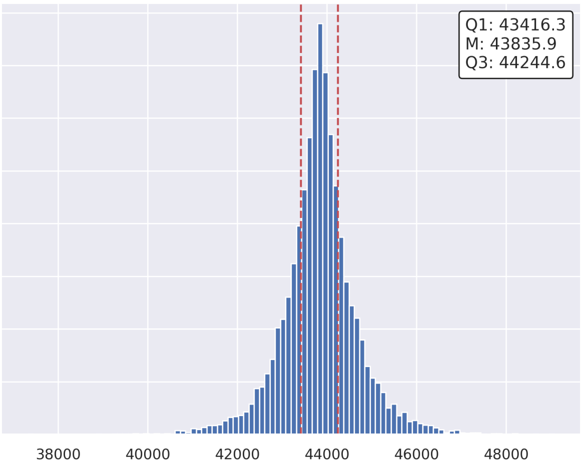


Figure 2.10: The distribution of differentially private query results for the sum of all incomes.

Sum queries A *sum query* calculates the sum over certain attributes (given by the function $a(\cdot)$) of all individuals $D^{(i)}$ which satisfy some condition (described by the indicator variable $\mathbb{I}(\cdot)$):

$$f_{sum}(D) = \sum_{i=1}^{|D|} a(D^{(i)}) * \mathbb{I}(D^{(i)}).$$

In contrast to the previous mechanisms, the influence of a single individual and thereby the sensitivity of the query can be much higher depending on the scenario. The query sensitivity Δf does fully depend on the attributes the sum is computed over. For a maximum individual influence C we get the sensitivity

$$\Delta f = C = \max_{D \in \mathcal{D}} a(D) - \min_{D \in \mathcal{D}} a(D).$$

By using the Laplace mechanism with random noise drawn from $Lap(\frac{C}{\varepsilon})$ we obtain a differentially private sum query. For sum queries with an unknown sensitivity, we can limit the values to a reasonable range $[0, \ldots, C]$ or $[-\frac{C}{2}, \ldots, \frac{C}{2}]$. Outliers are reduced to the lower or upper bound. This process is also referred to as *clamping*.

Coming back to our income example, we now want to compute a differentially sum over all incomes for a real value of $f_{sum}(D_{income}) = 43,841$. In comparison to the examples of counting and histograms, it is more complicated to determine the sensitivity of the function now. We have to choose reasonable minimum and maximum income values and compute the sensitivity of the query from these values. For our example we use 0 as the minimum and 150,000 €[26] as the maximum value and obtain a sensitivity $\Delta f = 150$. Therefore, we must add noise drawn from $Lap(\frac{150}{\varepsilon})$ with $\varepsilon = 0.5$ to the real result. Figure 2.10 shows the distribution of sum query results. Even though the absolute amount of required noise is much higher than in the query examples covered before, the relative result error is in a similar range since the sum result is much larger than the results covered before.

Average queries The following section is based on ideas provided by Brubaker and Prince [BP21]. An *average query* calculates the average over certain attributes (given by the function $a(\cdot)$) of all individuals which satisfy some condition (described by the indicator variable $\mathbb{I}(\cdot)$):

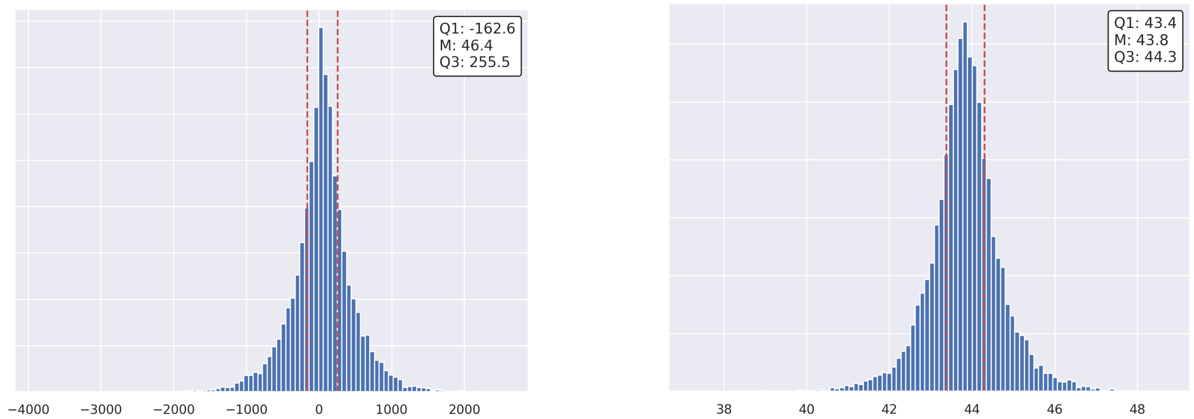
$$f_{avg}(D) = \frac{\sum_{i=1}^{|D|} a(D^{(i)}) * \mathbb{I}(D^{(i)})}{\sum_{i=1}^{|D|} \mathbb{I}(D^{(i)})}.$$

The easiest approach for computing a differentially private average consists in the direct application of the Laplace mechanism to the average function. Like for the sum query considered above, the sensitivity of the average query is formed by a to be determined value C, which depends on the specific scenario. This sensitivity takes the case, in which a single individual is responsible for the full average, into account. Therefore we need the same amount of noise as for sum queries. But this method has a big disadvantage with respect to the required amount of noise. While the final sum value in a sum query is generally much higher than the individual summands and the sensitivity of the query – which reduces the influence of the added noise – the average is generally smaller than the sensitivity of the query. Therefore, the required amount of added noise has a high impact on the result of the query. This can also be illustrated by the fact that the noise must also cover the worst case, which, as described, consists of a single person being included in the calculation.

One way to prevent this problem is the combination of the DP mechanisms for sum queries and counting queries. By employing the sequential composition property (see Section 2.7.2) of DP we can apply both mechanisms with privacy parameters ε_1 and ε_2 and $\varepsilon_1 + \varepsilon_2 = \varepsilon$. The result of this combined average query is obtained by dividing the result of the sum query by the result of the counting query for given privacy parameters. This approach preserves the privacy guarantees of ε-DP while simultaneously drastically reducing the influence of the added noise to the result of the query.

This relation is visualized in Figure 2.11 for our income dataset. The real income average is $f_{avg}(D_{Income}) \approx 43.8$. For the simple average mechanism depicted in Figure 2.11a we directly add noise drawn from $Lap(\frac{150}{\varepsilon})$ for $\varepsilon = 0.5$ (the same amount as for the sum

[26] This choice might exclude the income of the CEO of a multi-billion dollar company (who is almost always part of ordinary income datasets in data privacy research). However, these are exactly the decisions in DP which provide quite accurate query results while still preserving the privacy of CEOs.



(a) Simple average mechanism.

(b) Combined average mechanism.

Figure 2.11: A comparison between the simple and combined average mechanism.

query) to the real result. We can observe that the amount of required noise renders the results nearly meaningless, with income average quartiles ranging from -162.6 up to 255.5. In comparison, combining the sum query with noise drawn from $Lap(\frac{150}{\varepsilon_1})$ and the counting query with noise drawn from $Lap(\frac{1}{\varepsilon_2})$ and $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2} = 0.25$ provides accurate differentially private average results. These results are shown in Figure 2.11b, where we observe income average quartiles of 43.4 and 44.3.

Further differentially private mechanisms for computing averages, which provide varying results depending on the privacy parameter ε, are presented by Li et al. [Li+16].

Report Noisy Max

Another elegant mechanism, which – similar to computing averages – allows higher accuracy in comparison to multiple single queries, is called *report noisy max* [DR+14]. This mechanism allows us to compute the element r in a set of possible results $\mathcal{R}$ with the highest score given by a score function $u : \mathcal{D} \times \mathcal{R} \rightarrow \mathbb{R}$ for a database x:

$$f_{rnm}(D) = \operatorname*{argmax}_{r \in \mathcal{R}} u(D, r).$$

For example, the score function u can simply count the number of occurrences of elements in the database having the element r as an attribute, such as the type of cancer in a database of cancer patients. Report noisy max would allow us to find the prevalent cancer type in the database. But the mechanism also enables us to find the element r with the highest score in scenarios, in which an individual can contribute to the count of multiple elements $r \in \mathcal{R}$ (in comparison to histogram queries).

The basic idea of the mechanism is to add Laplace noise to the score for each element $r \in \mathcal{R}$ and just return the element with the highest score. The complete mechanism is shown in Listing 2.2.

Since the mechanism just returns a single element, we can achieve ε-DP, independently of the size of the result set $|\mathcal{R}|$. A proof for this statement is provided by Dwork, Roth,

```
function reportNoisyMax(D, u, $\varepsilon$)
    for $i \leftarrow \{1, \ldots, |\mathcal{R}|\}$
        $Y_i \leftarrow_R Lap(\frac{\Delta u}{\varepsilon})$
        $c_i \leftarrow u(D, r_i) + Y_i$
    noisyMaxIdx = $\operatorname{argmax}_{i \in \{1, \ldots, |\mathcal{R}|\}} c_i$
    return $\mathcal{R}$[noisyMaxIdx]
```

Listing 2.2: The report noisy max mechanism.

et al. [DR+14]. In comparison, we just achieve $|\mathcal{R}|\varepsilon$-DP when we perform the $|\mathcal{R}|$ queries independently and choose the highest noisy score value afterwards.

Exponential Mechanism

With *report noisy max* we already presented a mechanism, which allows non-numeric outputs. Another mechanism, which allows this, is the *exponential mechanism* [MT07]. In comparison to *Report Noisy Max*, this mechanism does not add noise to the results of the computation but to the probabilities for all outcomes. In this way we can answer queries in a reasonable manner, for which adding noise to single outcomes would not make sense. An example given by Dwork, Roth, et al. [DR+14] deals with an auction scenario in which individual bids are to be protected, but noise in individual bids can completely prevent sensible price determination and thus tamper with the outcome of the auction.

The main idea behind the exponential mechanism is based on a utility function $u : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$, which assigns a score value to each result r in the set of possible result $\mathcal{R}$. This score describes the quality of an answer as a result of the query – in the case of illnesses, for example, the frequency of an illness. The exponential mechanism simply consists in outputting a result value $r \in \mathcal{R}$ based on the probability distribution proportional to $\exp\left(\frac{\varepsilon u(D,r)}{2\Delta u}\right)$. The sensitivity Δu for the utility function is computed over all elements $r \in \mathcal{R}$ for databases D_1, D_2:

$$\Delta u = \max_{r \in \mathcal{R}} \max_{\|D_1 - D_2 \leq 1\|_1} |u(D_1, r) - u(D_2, r)|.$$

A proof for the fulfillment of ε-DP is provided by Dwork, Roth, et al. [DR+14].

2.7.4 (ε, δ)-Differential Privacy and the Gaussian Mechanism

In this section we cover a relaxation of ε-DP, called (ε, δ)-DP [Dwo+06a] (sometimes also referred to as *approximate DP*). The parameter δ can informally be interpreted as the probability that the guarantees of DP do not hold[27]. We start with the definition of (ε, δ)-DP and afterwards cover the Gaussian mechanism, which fulfills this definition of privacy.

[27] This interpretation is formally incorrect, but allows for an intuitive understanding of the parameter. Details on the real meaning of the parameter δ can be found in an excellent blog post by Desfontaines [Des20].

Definition 2.7.2. A randomized algorithm M with domain $\mathcal{D}$ is called (ε, δ)-differentially private, if for all S in the range of M and for all $D_1, D_2 \in \mathcal{D}$ with $||D_1 - D_2||_1 \leq 1$

$$Pr[M(D_1) \in S] \leq e^{\varepsilon} Pr[M(D_2) \in S] + \delta$$

holds. The probability space follows from the randomization of M.

We have seen the Laplace mechanism, which works by drawing noise from the Laplace distribution and fulfills ε-DP. Another DP mechanism, the so-called *Gaussian mechanism*, works by drawing noise from the normal (or Gauss) distribution. For the Gaussian mechanism we can use a variant of the sensitivity, which just differs in the used distance measure:

$$\Delta_2 f = \max_{\substack{D_1, D_2 \in \mathcal{D} \\ \|D_1 - D_2\|_1 = 1}} \|f(D_1) - f(D_2)\|_2 .$$

$\|\cdot\|_2$ describes the $l2$-norm (also known as the *Euclidean distance*) of the function results. For the Gaussian mechanism we compute

$$\mathcal{M}_{\text{Gauss}}(D, f(\cdot), \varepsilon, \delta) = f(D) + (Y_1, \ldots, Y_k),$$

in which Y_i represents a random variable drawn from the normal distribution $\mathcal{N}$ with mean $\mu = 0$ and variance $\sigma^2 = \frac{2\ln(1.25/\delta)\cdot(\Delta_2 f)^2}{\varepsilon^2}$[28]. The Gaussian mechanism fulfills (ε, δ)-DP. A proof for this statement is provided by Dwork, Roth, et al. [DR+14]. (ε, δ)-DP provides similar composition properties like ε-DP [DR+14].

The question remains as to why or when the Gaussian mechanism and (ε, δ)-DP should be used, even though it provides less strict privacy guarantees in comparison to ε-DP. A first reason is the familiarity of data analysts with the normal distribution and its properties compared to the less often used Laplace distribution. Another, more important reason shows up when we look at multiple queries being performed on the same database. The amount of required noise to achieve the same level of privacy with respect to ε generally is lower for the Gaussian mechanism when performing multiple queries (at the cost of $\delta > 0$). This is especially relevant for differentially private machine learning applications, in which noise has to be applied to large vectors (see Section 2.7.6).

2.7.5 Choosing Parameters ε and δ

When deploying DP into a practical setting, we have to look at instantiations of the privacy parameters – or in other words: How do we choose ε and δ? They play a vital role in the privacy protection capabilities of a system and decide between strong and meaningless protection of privacy [DKM19]. The importance of choosing appropriate parameters can be compared to the choice of security parameters in cryptosystems: The key length establishes the security of the crypto system and using a short key can compromise the security of an otherwise secure system. But while choosing large keys

[28] It has been shown, that the original formulation of the mechanism given here adds more noise than necessary. Balle and Wang [BW18] provide details on how to choose the standard deviation σ for tighter bounds.

can at most interfere with the performance of a cryptosystem, choosing oversized privacy parameters can impair the quality of the query result and in the worst case render it meaningless. Furthermore, in comparison to the choice of key length, for which suitable parameters are quite easily determinable, the choice of privacy parameters is not directly obvious and can depend on properties of the scenario: the value of data usage, the sensitivity of the data, present data attributes, and practices and policies to restrict the analysts who query the data [DKM19]. To some extent it is a societal problem to choose the right degree of privacy protection in specific scenarios. But even an informed decision about this tradeoff does not allow to directly determine an appropriate value of ε. A similar argument is given by Lee and Clifton [LC11]. According to them, a suitable value for ε is not obvious due to the missing direct relation between this privacy parameter and practical disclosure risks as given in Section 2.1.

A survey conducted by Dwork, Kohli, and Mulligan [DKM19] examined how the parameter ε was chosen in practice. They found a wide range of methods:

- Simulations were performed to find a value of ε which did disturb the query result little enough to still meet the business requirements.
- Others performed threat modeling to deduce suitable parameter values from the resulting models.
- Some practitioners fell back on values being used in existing implementations.
- A final group admitted to use somewhat arbitrary choices without much consideration.

Based on these results, Dwork, Kohli, and Mulligan propose the idea of a *Epsilon Registry* including amongst others the paths of privacy loss, the choice of ε and the variant of DP, and a justification of implementation details. This registry could serve as a resource for shared learning in the field of DP. Furthermore, the disclosure of a company's choice of ε and other details would allow regulators to assess the effect of used privacy measures.

Since such a registry currently does not exist, we present several existing approaches in this chapter which can support adopters of DP in the choice of the privacy parameter ε.

Visual Support Tools

There are several publications providing visual tools which allow the data analyst to directly see the influence of their choices of privacy parameters. Gaboardi et al. [Gab+18] developed the tool *PSI*[29] which allows sharing and exploring privacy-sensitive datasets in a differentially private fashion. Hay et al. [Hay+16] provide *DPComp*[30], a tool to visually assess the influence of DP on the accuracy of different algorithms on publicly available datasets. *Overlook*[31], developed by Thaker et al. [Tha+20], enables interactive, differentially private exploration of data and the impact of privacy parameters. John et al. [Joh+21] present *differential privacy policy tool (DPP)*[32] which abstracts from

[29] A deployed version of *PSI* is available at `http://psiprivacy.org/` (visited on 23.11.2022), the code at `https://github.com/opendp/PSI` (visited on 23.11.2022).

[30] *DPComp* is accessible at `https://www.dpcomp.org/` (visited on 17.11.2022).

[31] The code of *Overlook* is available at `https://github.com/vmware/hillview` (visited on 23.11.2022).

[32] Unfortunately, we are not aware of an available deployment of this tool.

concrete parameters and allows data analysts to explore the relation between risk, sensitivity and trust w.r.t added noise with simple parameter choices from very low to very high. Nanayakkara et al. [Nan+22] introduce *Visualizing Privacy (ViP)*[33] which depicts expected accuracy and privacy risks based on different choices for ε.

An Economic Model

Hsu et al. [Hsu+14] propose an economic model for choosing the privacy parameters of DP in privacy-preserving studies. The main idea of their approach is to assign costs to all possible privacy violation events. They look at the problem from two sides: the side of the *data analyst* and the side of a potential *participant* in the study. The data analyst has a minimum required accuracy α given by an accuracy function $A(\varepsilon, N)$ for the study and a budget B they can spend on the study to compensate participants for possible disadvantageous events arising from their participation. The participant decides to participate in the study only if their compensation exceeds the increase in the expected worst-case cost C in comparison to the costs of not participating E. The data analyst can then compute suitable values for ε by respecting constraints with respect to budget and accuracy. Additionally, the authors provide several refinements for their model in terms of additional constraints for upper and lower bounds on ε and in terms of using (ε, δ)-DP. Even though this approach and the introduction of several additional parameters might seem complex, Hsu et al. argue that their model makes real-world considerations explicit which are condensed in the single parameter ε in the original definition of DP. On the other hand, they also admit that the assumption of participants being able to estimate the expected cost of events when they participate as well as when they do not participate might be a too simple approach for some applications.

A Vote-based Model

The approach of Kohli and Laskowski [KL18] differs from the other approaches covered in this section in that they empower the individuals to choose the value of ε themselves. According to the authors this has several benefits, including the ideas that individuals would better understand their own privacy risks in specific situations than uninvolved data analysts, and that it would allow individuals to partially control the behavior of a system directly affecting their lives. The main idea is to let each individual vote for their preferred value of ε (potentially in a set of appropriate values $\varepsilon_1, \ldots, \varepsilon_k$). A chooser mechanism aggregates these values and outputs the final choice for $varepsilon$. This mechanism has to fulfill three properties: It has to be *truthful*, meaning that the mechanism should provide individuals an incentive to report their real preference. It must keep votes *private*, as the sole information about an individuals privacy preferences can already indicate the motivation to hide something. Finally, it should be *anonymous* (in a game-theoretic sense), meaning that it does not favor the vote of some individual over the vote of another one. The authors provide different mechanisms for two scenarios: arbitrary preferences and single-peak preferences – a variant in which a single individual has high privacy requirements in comparison to others.

[33] A demo can be found at `https://priyakalot.github.io/ViP-demo/` (visited on 17.11.2022).

The central assumption of this approach (and possibly a severe limitation) is that individuals are able to understand their personal privacy risks and to accurately quantify these in terms of ε. As covered in the beginning of this section, there seems to be now agreement even under privacy experts on how to choose these parameter. It is unclear, how each directly affected individual should be able to do so.

A Model based on Estimation Theory

Naldi and D'Acquisto [ND15] provide an approach for a more intuitive choice of privacy parameters when using Laplace noise for counting queries based on interval estimation. The original Laplace mechanism just uses ε as a privacy parameter, which determines the scale of the Laplace distribution noise is drawn from. Naldi and D'Acquisto describe this as an unintuitive way of determining the desired level of privacy. Instead, they propose the usage of two parameters, namely the *confidence interval width* w and the *confidence level* p, which describe the probability that the real result c of the query is placed in the given interval around the differentially private result. Given these parameters, the required value for ε can be computed as

$$\varepsilon = -\frac{\ln(1-p)}{wc}.$$

The authors provide an interesting idea which might prove useful for a more intuitive way of choosing ε. A major disadvantage is the dependence of ε on the real result of the query c, which prevents weighting privacy and utility before performing the query. Furthermore, it is not clear how to translate their results to other mechanisms.

A Risk-based Model

Lee and Clifton [LC11] provide a way to compute a suitable value for ε based on a specific adversary model: the probabilistic disclosure of the presence of an individual in the dataset. Their approach uses an adversaries posterior belief about the presence of the individual. This is related to our notion of membership disclosure (see Section 2.1). Providing an upper tolerable bound for this belief allows for computing a suitable parameter ε. The computation is based on the Laplace mechanism (see Section 2.7.3) and incorporates the sensitivity of the desired function Δf, the maximum distance between function values of all possible worlds Δv and the number of elements in these worlds n – so it is highly application and data specific. Using these values they propose the following choice for ε given the upper bound for the adversaries posterior belief p:

$$\varepsilon \leq \frac{\Delta f}{\Delta v} \ln \frac{(n-1)p}{1-p}.$$

Mehner, Voigt, and Tschorsch [MVT21] extend the results of Lee and Clifton. By just considering worst-case estimates of the number of elements and the sensitivity, they introduce the *global privacy risk* and *leak*, which allow for global upper bounds. Furthermore, they apply these risks to the randomized response mechanism (see Section 2.7.3) for a more intuitive explanation of privacy risks.

Another, more general Bayesian interpretation of DP, which we describe in the next section, allows for more general statements about choosing ε.

Bayesian Interpretation

The following interpretation comes from [Des18] and is based on [KS14a]. It allows for a graphic view on DP and especially the privacy parameter ε. The interpretation follows from looking at the information which is available to an adversary. Given a database D, a ε differentially private algorithm A and an algorithm output O, resulting from applying the algorithm A to the database D. Depending on the output O the adversary's beliefs about the presence or absence of an individual in database D can change. We can compute lower and upper bounds for the maximum changes of these adversary's beliefs.

We assume the adversary to possess knowledge about all individuals in the database with the exception of a single individual. For this individual the adversary does not know whether the individual is part of the database. We call this an adversary with full background knowledge. $\mathbb{P}[D = D_{in}]$ describes the initial beliefs of the adversary (in terms of a probability) that the individual is part of the database. Accordingly, $\mathbb{P}[D = D_{out}] = 1 - \mathbb{P}[D = D_{in}]$ describes the converse probability for this event. When the adversary learns the output O, they can update their beliefs depending on this output. We can model this updated beliefs as the conditional probability $\mathbb{P}[D = D_{in} \mid A(D) = O]$. The privacy guarantees of ε-DP allow us to quantify the maximum changes to the adversary's beliefs. For this purpose we employ Bayes' Theorem

$$\mathbb{P}[D = D_{in} \mid A(D) = O] = \frac{\mathbb{P}[D = D_{in}] \cdot \mathbb{P}[A(D) = O \mid D = D_{in}]}{\mathbb{P}[A(D) = O]}.$$

For better readability, we can replace $\mathbb{P}[A(D) = O \mid D = D_{in}]$ with $\mathbb{P}[A(D_{in}) = O]$. To get rid of the unknown term $\mathbb{P}[A(D) = O]$ we instead consider the quotient of updated beliefs $\mathbb{P}[D = D_{in} \mid A(D) = O]$ and $\mathbb{P}[D = D_{out} \mid A(D) = O]$:

$$\frac{\mathbb{P}[D = D_{in} \mid A(D) = O]}{\mathbb{P}[D = D_{out} \mid A(D) = O]} = \frac{\mathbb{P}[D = D_{in}]}{\mathbb{P}[D = D_{out}]} \cdot \frac{\mathbb{P}[A(D_{in}) = O]}{\mathbb{P}[A(D_{out}) = O]}.$$

The ratio between $\mathbb{P}[A(D_{in}) = O]$ and $\mathbb{P}[A(D_{out}) = O]$ is directly present in the definition of ε-DP and is bounded by the privacy parameter:

$$e^{-\varepsilon} \leq \frac{\mathbb{P}[A(D_{in}) = O]}{\mathbb{P}[A(D_{out}) = O]} \leq e^{\varepsilon}.$$

This leads to the inequality

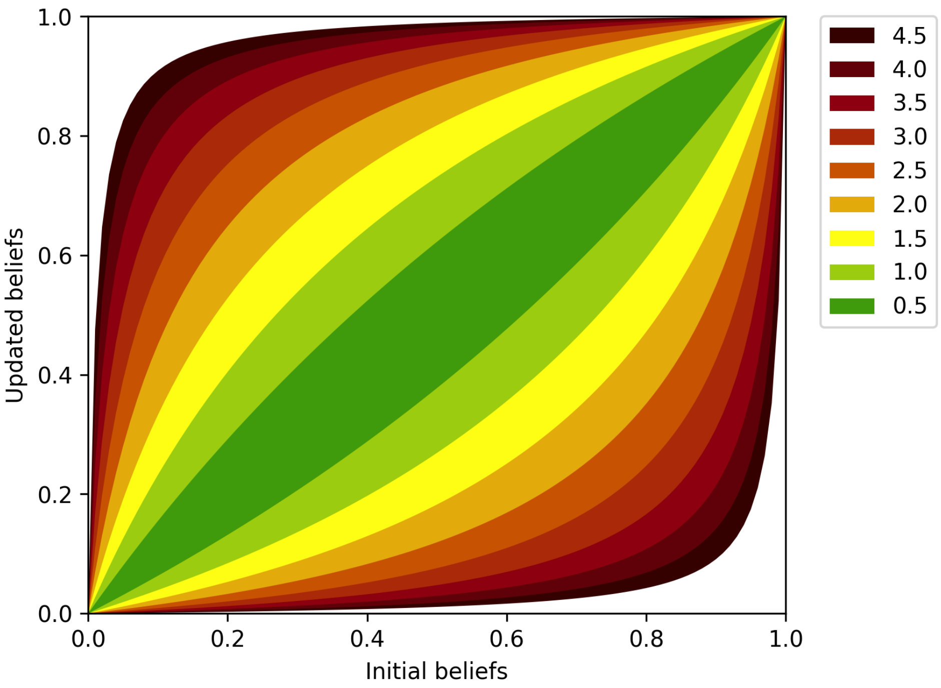


Figure 2.12: Relationship between an adversary's initial and updated beliefs for different values of the privacy parameter ε. Own illustration based on [Des18].

$$e^{-\varepsilon} \cdot \frac{\mathbb{P}[D = D_{in}]}{\mathbb{P}[D = D_{out}]} \leq \frac{\mathbb{P}[D = D_{in} \mid A(D) = O]}{\mathbb{P}[D = D_{out} \mid A(D) = O]} \leq e^{\varepsilon} \cdot \frac{\mathbb{P}[D = D_{in}]}{\mathbb{P}[D = D_{out}]}.$$

By replacing $\mathbb{P}[D = D_{out}]$ with $1 - \mathbb{P}[D = D_{in}]$ and $\mathbb{P}[D = D_{out} \mid A(D) = O]$ with $1 - \mathbb{P}[D = D_{in} \mid A(D) = O]$ we obtain

$$e^{-\varepsilon} \cdot \frac{\mathbb{P}[D = D_{in}]}{1 - \mathbb{P}[D = D_{in}]} \leq \frac{\mathbb{P}[D = D_{in} \mid A(D) = O]}{1 - \mathbb{P}[D = D_{in} \mid A(D) = O]} \leq e^{\varepsilon} \cdot \frac{\mathbb{P}[D = D_{in}]}{1 - \mathbb{P}[D = D_{in}]}.$$

Solving the inequality for $\mathbb{P}[D = D_{in} \mid A(D) = O]$ results in

$$\frac{\mathbb{P}[D = D_{in}]}{e^{\varepsilon} + (1 - e^{\varepsilon}) \cdot \mathbb{P}[D = D_{in}]} \leq \mathbb{P}[D = D_{in} \mid A(D) = O] \leq \frac{e^{\varepsilon} \cdot \mathbb{P}[D = D_{in}]}{1 + (e^{\varepsilon} - 1) \cdot \mathbb{P}[D = D_{in}]}.$$

Figure 2.12 visualizes this inequality for different values of the privacy parameter ε. This allows us to assess the worst-case changes of the adversary's beliefs caused by some output O. An example conclusion we can draw from it is the following: When we use a privacy parameter $\varepsilon \geq 2,5$ the result of a ε-differentially private algorithm can already convince an adversary without prior knowledge ($\mathbb{P}[D = D_{in}] = 0,5$) of the existence of an individual in the database in the worst case ($\mathbb{P}[D = D_{in} \mid A(D) = O] \geq 0,9$). Similar statements can be used for the purpose of risk assessment when choosing the privacy parameter ε.

2.7.6 Outlook on Further Topics

So far we provided an introduction to DP and its most relevant basic concepts. However, there are multiple directions in which research extended these basic concepts or applied DP in other domains. In this section we outline some of these as a starting point for further research. This includes the concept of centralized, local and hybrid DP, adjusted DP definitions, using DP in the field of ML, and the differentially private generation of synthetic data.

Centralized, Local, and Hybrid Differential Privacy

Up to this point we have dealt especially with what is referred to as *centralized DP* (also referred to as *trusted curator model* [Ave+17] or *global model* [DP20]): A central trusted party is in possession of the complete dataset, performs some differentially private computation and can publish the result without putting individuals in the dataset at risk.

Another possibility which removes the need for a trusted centralized party is called *local DP*, introduced by Kasiviswanathan et al. [Kas+11]. In local DP the differentially private mechanism is performed by each user locally and the perturbed results are sent to a central party. This removes the risk of data disclosure by an adversarial central party so that the central party does not have to be trustworthy anymore[34].

Local DP requires a slightly changed definition in comparison to ε-DP (see Section 2.7.1). The following definition is based on the one provided by Cormode et al. [Cor+18].

Definition 2.7.3. An algorithm M satisfies ε-*local DP*, where $\varepsilon \geq 0$, if and only if for all y in the range of M and for any input v and v', we have

$$\forall y \in Range(M) : \mathbb{P}[M(v) = y] \leq e^{\varepsilon}\mathbb{P}[M(v') = y].$$

The algorithm M is applied to each individual's data independently. Informally, this means that local DP bounds the differences of result probabilities for each individual.

Most local differentially private mechanisms are based on randomized response, but the Laplace or Gaussian mechanism can be used as well(see Section 2.7.3). The downside of this approach is that it requires much noise to be added to the individual's data when a reasonable value for ε is chosen. Due to this requirement local DP is only suitable for scenarios with many individuals participating in the process.

Local DP has attracted a lot of attention in recent years and is used by Google [EPK14], Microsoft [DKY17], and Apple [Tea17]. Yang et al. [Yan+20] provide an extensive survey and Cormode et al. [Cor+18] a short tutorial on local DP.

But due to the non-universal applicability of local DP, researchers have looked for solutions in the space between central DP with the requirement of a trusted central party and local DP which imposes large amounts of noise to the computations. There are two

[34] At least the party has not to be trustworthy with respect to preserving the privacy of individuals. Trustworthiness regarding the publication of correct computation results is still needed.

general directions for this middle ground: distributed DP and hybrid models Kairouz et al. [Kai+21]. In *distributed DP* a secure computation function, for example, based on SMPC, is employed to aggregate the client results before sending the differentially private aggregation result to the server. One example is provided by Dwork et al. [Dwo+06a], who provide a solution for distributedly compute Gaussian noise. Another example is the Encode-Shuffle-Analyze framework and its implementation *Prochlo* introduced by Bittau et al. [Bit+17]. *Hybrid models*, on the other hand, allow individuals to choose between placing trust in the central party and using local DP and provide a way to combine the received results. An example of a hybrid approach is *BLENDER* introduced by [Ave+17].

Other Semantic Privacy Models

In previous sections we have covered two definitions of DP: ε-DP and (ε, δ)-DP. But in the literature a large number of further definitions can be found which relax, change or generalize different aspects of the original definition. Examples include Rényi DP [Mir17], personalized DP [JYC15], noiseless privacy [Bha+11], Pufferfish privacy [KM12], and coupled worlds privacy [Bas+13].

Desfontaines and Pejó [DP20] have provided a comprehensive survey for definitions and introduce a systematic taxonomy. They present seven dimensions in which the original DP definition can be modified: quantification of privacy loss, neighborhood definition, variation of privacy loss, background knowledge, formalism change, relativization of knowledge gain, and computational power. Furthermore, they categorize about 200 definitions found in the literature according to these dimensions and their relation to the original definition and among each other.

Differentially Private Machine Learning

One field in which DP has attracted high attention is the field of ML. Adversaries with access to the model parameters or even just with the capability to query the model are potentially able to execute a variety of attacks, amongst others:

- *Membership inference attacks* [Sho+17; NSH19] determine if a data record was part of the training data.
- *Attribute inference attacks* [Yeo+18] infer missing information about an incomplete data record.
- *Model inversion attacks* [FJR15] attempt to reconstruct training data records.

These attacks can represent risks to individuals in the training datasets when dealing with sensitive (for example, medical) data and DP can be used to reduce these risks. While early results using DP for simple models, such as regression models [Zha+12a] and decision trees [FS10], received considerable attention, the combination of DP and ML received a boost with the work of Abadi et al. [Aba+16]. They provided a way to introduce DP into *deep learning* – training neural networks with multiple layers which are suitable for a variety of problems, including image recognition [He+16], natural language processing [OMK21], or even board games [Dee24]. DP is especially relevant for these types of models since the complexity of trained models can encode details of data records from training data and leak this data to adversaries. DP allows to reduce the impact a single individual has to the training process and the resulting model and can proof helpful in reducing the success probability of these attacks.

Recent surveys in this area are provided by Ouadrhiri and Abdelhadi [OA22] (focuses on deep learning), Fletcher and Islam [FI19] (deals with decision trees), and Liu et al. [Liu+21] (considers the interplay between privacy and ML in different areas).

Jayaraman and Evans [JE19] looked at the influence of the DP privacy parameters and showed that the privacy-utility tradeoff (see Section 2.2) unsurprisingly also applies to DP when used in ML.

A subfield of ML in which DP plays an important role is federated learning (FL), introduced by McMahan et al. [McM+17]. FL describes a concept for distributed machine learning in which the training is performed iteratively by the participants on their local data without sharing this data with other parties. The local updates are aggregated in a secure manner and finally result in a trained model. DP can be used to reduce the risk of disclosure via local updates and also reduce the influence of a single individual on the final model. Recent surveys about FL which also look into DP are provided by Kairouz et al. [Kai+21], Li et al. [Li+20], Yang et al. [Yan+19], Rieke et al. [Rie+20], and Wei et al. [Wei+20].

Differentially Private Synthetic Data Generation

Further mechanisms try to bridge the gap between differentially private queries and microdata release by *synthetic data generation*. Instead of publishing an altered version of the original data, a new synthetic dataset is constructed in a differentially private manner, which statistically resembles the original data. We briefly mention some publications in this area as a starting point for further results. Machanavajjhala et al. [Mac+08] look into synthetic data generation for commuting patterns under a relaxed definition of DP. Hardt, Ligett, and Mcsherry [HLM12] present *MWEM* which produces synthetic datasets that respect any set of linear queries. Blum, Ligett, and Roth [BLR13] provide methods for releasing synthetic datasets for answering particular classes of queries, including counting and interval queries. Qardaji, Yang, and Li [QYL14] develop a method for differentially private synthetic marginal tables. Ping, Stoyanovich, and Howe [PSH17] provide *DataSynthesizer*, an open-source tool for creating differentially private datasets from sensitive data. Zhang et al. [Zha+17] present *PrivBayes*, a synthetic data generation method based on Bayesian networks with a focus on high-dimensional data. Torkzadehmahani, Kairouz, and Paten [TKP19] use a differentially private version of a conditional GAN for generating artificial image data. Jiang, Zhou, and Grossklags [JZG22] translate the problem to the distributed setting and use local DP, FL, and generative autoencoders to generate synthetic data. An overview of further approaches to differentially private synthetic data generation and a comparison of the utility of resulting datasets is given by Bowen and Liu [BL20]. A recent survey is provided by Hu et al. [Hu+24]. The differentially private generation of synthetic data is a field of active research and new results are constantly published, for example, with a focus on increasing the accuracy or performance of models for specific tasks [LVW21; Vie+22; Ayd+21].

2.7.7 Concluding Remarks

As already covered in Section 2.7, the semantic privacy model DP entails several useful properties [NHF15]. It protects individual information in worst-case scenarios and does not require any assumptions with respect to potential attacks, in other words, the (proven and quantifiable) guarantees are independent of any auxiliary information, adversary's capabilities, and currently known or future attacks techniques. It is possible to directly quantify the tradeoff between privacy and utility (see Section 2.2) via the privacy parameter ε. Furthermore, we have seen that DP provides composability – the possibility to perform multiple computations in a way that still allows for the quantification of privacy loss (see Section 2.7.2). Hardt [Har15] makes the point that in comparison to other models, such as the de-identification measures required by the HIPAA safe harbor mechanism (see Section 2.3.2), DP can even provide better data utility because the original data distribution remains unchanged. Additionally, they argue that DP provides a sort of stability guarantee for ML applications since the idea of learning about populations not specific individuals is fundamental in statistical learning. Aaron Roth, one of the leading minds behind DP, mentions that the main benefit of DP is increased access to data in scenarios in which data access is otherwise problematic due to regulations [Har15]. One example for this is the collection of large-scale user data with the help of *RAPPOR* by Google [EPK14] – data which wasn't accessible at all beforehand.

On the other hand, DP is not a silver bullet for each and every problem in privacy-preserving data publishing and mining. First, the main idea of DP is to bound the influence a single individual's data has on the result of a computation to protect their privacy. Therefore DP is not suitable for studying outliers or small datasets and it might hide important specifics of small subpopulations in larger datasets [DKM19]. In comparison to simple de-identification techniques or syntactic privacy models, which result in modified datasets, there is no one-to-one correspondence in DP results [Coh22], which might be expected by practitioners. Additionally, the idea of adding noise to the original data can create opposition from practitioners [CT13], ignoring the fact that the original data itself can often be noisy [Des21]. Therefore, providing confidence intervals with DP results is a quite important method to increase the interpretability of results [Gue+20]. Another thing to be aware of is that DP does not prevent conclusions to be drawn about individuals based on knowledge derived from differentially private computations [DKM19; Nis+17]. Correlations like *smoking increases the risk of lung cancer* learned in a differentially private way can still cause harm to a smoking individual, for example, in the form of higher insurance fees. But DP guarantees, that this harm does not depend on the presence of the individual in the database (see also the discussion about *population disclosure* in Section 2.1).

Furthermore, there is a variety of challenges for the practical application of DP. The first challenge lies in the correct choice for privacy parameters, which is by no means a simple task (see Section 2.7.5), amongst others, because there is no clear relationship between the parameter ε and legal concepts like identifiability [CT13] (cf. Section 2.3). With regard to these difficulties, Dankar and El Emam [DE13] also mention the hurdles that can arise when it comes to explaining parameter choices to individuals or justifying them in potential litigation. Even when a suitable value for ε is chosen, there exist further related challenges [CT13; DE13]. When ε is treated as a *privacy budget*, practitioners have to think about how to split this budget between multiple users. Additionally, they must decide how much budget to spend for different queries, which is especially challenging as the amount required to achieve a query result deemed sufficiently accurate can vary dramatically between different query types. These questions are especially challenging in scenarios, in which potential users and query types are not known in advance. Domingo-Ferrer, Sánchez, and Blanco-Justicia [DSB21] mention the danger of using unreasonably large values for ε, in which case there are no useful privacy guarantees provided by DP anymore. They further criticize some simplifications regarding composability found in practice. For example, Apple uses the full privacy budget of their DP solution for each individual per day, so that this budget is violated due to sequential composition (see Section 2.7.2).

We have already mentioned the need to illustrate the effect of parameter choices. But there is also the necessity to explain DP to practitioners as well as affected individuals and to clarify potential misconceptions [Gue+20]. Anecdotal evidence for this necessity can be found in the article *Fool's Gold: An Illustrated Critique of Differential Privacy* [BMS13]. This article confidently presented some factually incorrect arguments against the use of DP and produced several harsh criticisms from researchers engaged in DP[35]. There is a line of research dealing with the question of how to make DP more understandable [Fra+22; KAF22; CKR21; Bul+17].

Further challenges for translating the theoretical DP results into practice include a lack of robust and usable implementations of common DP algorithms[36] [Har15], missing domain-specific tutorials for the application of DP [Har15], the complexity of determining global sensitivities, especially in the multidimensional space [CT13], and the difficulties in securely implementing these results while preventing information leaks, such as the one caused by insecure floating-point number handling [Gue+20; Mir12].

After examining specific protection measures and their strengths and weaknesses in the last sections, in the next section we turn to practical examples where an inappropriate use of these measures for the processing of personal data has led to publicized datasets incorporating a large re-identification risk for individuals.

2.8 Privacy Breaches

In the following we review some of the most relevant breaches of published "anonymized" datasets. Similar overviews of or references to these and other breaches are provided in several publication [El 13; DKM19; RHM19; Zig+20; SOT22].

[35] The interested reader can enjoy criticisms by Frank McSherry (https://github.com/frankmcsherry/blog/blob/master/posts/2016-02-03.md), Anand Sarwate (https://ergodicity.net/2014/11/03/an-exercise-in-uncareful-misreading/), and Adam Smith (https://web.archive.org/web/20180803034501/https://plus.google.com/+ShriramKrishnamurthi/posts/j2GtfgKAh6d), a response by Jane Bambauer (https://blogs.harvard.edu/infolaw/2016/05/17/diffensive-privacy/), and a response to the response by McSherry again (https://github.com/frankmcsherry/blog/blob/master/posts/2016-05-19.md) (visited on 14.06.2023).

[36] This challenge is currently approached by various parties. For a survey of existing DP frameworks see Section 6.4.3.

2.8.1 Chicago Homicide Victims Dataset

A study performed at the *Massachusetts Institute of Technology* in 2001 [Och+01] looked at a de-identified dataset of homicide victims in Chicago. By combining this dataset with the *Social Security Death Index* – a dataset provided by the Social Security Administration containing information about reported deaths, including individuals names and dates of birth – the authors were able to re-identify about 35 % of victims. Since the homicide dataset also contained details about the involvement of drugs, child abuse, gang violence, or domestic abuse in the homicide, there were clear implications to the privacy of the victims with potential bad consequences at least for their families.

2.8.2 Massachusetts Health Dataset

Sweeney [Swe97; Swe00; Swe15] was able to identify individuals in a dataset published by the *Group Insurance Commission*, a health insurance company for state employees. She combined the dataset with a voter registration list of Cambridge (Massachusetts) by taking advantage of the fact that both datasets contained individual's zip code, date of birth, and sex. This enabled her to exemplarily identify the Massachusetts governor's health records. Based on these insights, she developed k-anonymity as a measure against these types of linkage attacks (see Section 2.6.2).

2.8.3 AOL Dataset

In 2006, *AOL* published a dataset of 20.000.000 search queries from about 650.000 Americans stripped from directly identifying information [BZH06]. Reporters from the *New York Times* were able to identify an individual by their search queries, which included, for example, the last name of relatives or information about their city. The article mentions several sensitive search query topics, including sexual preferences, emotional state, and medical conditions.

2.8.4 Netflix Prize Dataset

Narayanan and Shmatikov [NS08] provide re-identification algorithms focusing on high-dimensional, sparse microdata, such as movie or book preferences or transaction histories, often used for *collaborative filtering*[37]. The algorithms are based on the idea that single individuals in a sparse dataset can be identified by a rather small number of non-null entries, such as movie ratings. This is especially true if these entries are rare in the dataset, for example, when an individual often watches non-mainstream movies.

They use the developed algorithm to identify individuals in the *Netflix Prize dataset*. The dataset was published in the context of a contest to improve the Netflix recommendation algorithm and consisted of about 100.000.000 movie ratings created by nearly 500.000 users (a fraction of about $\frac{1}{8}$ of all Netflix users). By correlating the dataset with background knowledge in the form of public *Internet Movie Database* data, the authors

[37] Also known as *recommender systems* or "Customers who bought this item also bought...".

were able to uniquely identify individuals. Although some consider their movie ratings as insensitive, the authors argue that sensitive information such as political views or sexual orientation can often be inferred from these ratings.

2.8.5 Social Networks Datasets

Narayanan and Shmatikov [NS09] look at the problem of de-anonymizing social network datasets, in which identifiers are stripped and just the structure of the network graph is preserved: individuals are represented as nodes and relationships between them, such as *follow* information in social networks, as edges in the graph. The authors show that this structural information is enough to re-identify individuals using auxiliary graphs and mapping algorithms on these graphs. They demonstrate the applicability of their algorithm on a de-identified Twitter dataset using a Flickr dataset as auxiliary information.

2.8.6 Kaggle Social Network Challenge Dataset

In 2011 Narayanan, Shi, and Rubinstein [NSR11] won the *Kaggle Social Network Challenge*, in which links between individuals in a large dataset of Flickr users stripped from user identities should be predicted, based on their ideas from [NS09]. They gamed the competition by de-anonymizing the dataset to a large extent using a combination of crawling public Flickr information and weighted graph matching. By combining the de-anonymized part of the dataset with a regular ML approach for the remaining data records, which they were not able to de-anonymize unambiguously, they achieved the best prediction results of all competitors.

2.8.7 Mobile Phone Location Dataset

De Montjoye et al. [De +13] use a dataset consisting of location data from 1.500.000 mobile phone users collected over the period of 15 months. The data records were stripped from directly identifying information. Nonetheless, the authors show that the uniqueness of mobility traces is high and that little background knowledge is required to re-identify individuals in such datasets.

2.8.8 Credit Card Dataset

Montjoye et al. [Mon+15] examine the credit card transactions (pseudonym, day of transaction, price) from 1.100.000 users collected in 10.000 shops over 3 months. The dataset was stripped from directly identifying information. They show that with just 4 known transactions, an adversary would be able to re-identify over 90 % of all individuals. Similar results are expected for similar sparse, high-dimensional datasets, such as “browsing history, financial records, and transportation and mobility data” [Mon+15].

2.8.9 New York Taxi Dataset

The New York *Taxi and Limousin Commission* publicly released a dataset of all taxi rides in New York including pickup and drop-off locations and times as well as fare and tip. This data has been abused to deduce residence and tipping patterns of celebrities by matching the rides data with public images of them getting in a taxi [Dou+16]. Furthermore, also the privacy of the taxi drivers could be violated due to bad hashing of the taxi license numbers.

2.8.10 Australian Health Dataset

Culnane, Rubinstein, and Teague [CRT17] exposed severe weaknesses in a dataset containing individual medical billing information about 2.9 million Australian citizens published by the federal Department of Health. While the patients' direct identifiers were obfuscated and only the gender and year of birth were provided, the dataset contained individual billing information including codes for the respective treatment or drug prescription, the slightly altered event date, the price, and the state in which the event happened. The authors show that this information is enough to uniquely identify specific individuals due to the sparsity of the dataset (see Section 2.8.4). They provide several graphic re-identification examples, including mothers identified by their children's date of birth, football players identified by required surgical treatments, and politicians identified by hospital admissions reported in newspapers. All these examples only required auxiliary information that was perceived as public information. Furthermore, the authors discuss implications for individuals when linking the dataset with several non-public datasets, such as the data of credit card companies.

2.8.11 edX Dataset

Cohen [Coh22] presents a re-identification attack against the *Harvard-MIT edX*[38] *dataset*, which contains student's demographic information, information about their activities, as well as their edX course outcomes. Even though the dataset has been "'properly de-identified' by 'statistical experts'" [Coh22] in a k-anonymous way, he presents three attacker models with varying potential background knowledge (prospective employer, casual acquaintance, edX classmate) and provides the amount of uniquely linkable data records for each of these models. Furthermore, by using public LinkedIn profile information he exemplarily identifies three students who failed at least one course with high certainty.

2.8.12 Reconstruction Attack against Aggregated Statistics

Recently, Dick et al. [Dic+23] show that publishing precise aggregate statistics derived from a private dataset D imposes the risk of being vulnerable to a specific kind of

[38] *edX* provides open online courses in a variety of disciplines accessible at `https://www.edx.org/` (visited on 26.04.2023) and was initially created by the Massachusetts Institute of Technology and Harvard University.

reconstruction attack. The introduced attack results in a set of potential data records ranked by the likelihood of their presence in the private dataset D. The attack is based on a recent result from synthetic data generation (see Section 2.5.9), called relaxed adaptive projection (RAP) [Ayd+21]. RAP is a randomized algorithm that takes as input a set of counting queries and their results on dataset D and generates a synthetic dataset D' by trying to solve the optimization problem of minimizing the difference between the counting query results on D and D'. The idea behind the reconstruction attack is to generate K synthetic datasets via RAP and output their union (with multiplicities) as a confidence set (ranked by number of occurrences). Due to the probabilistic nature of RAP, different result sets are generated. The intuitive idea behind the attack is that data records being constructed more often in these datasets are more likely to be present in D.

They evaluate the effectiveness of their reconstruction attack against synthetic microdata and real datasets released by the US Census Bureau. Their results indicate a reconstruction of a subset of all data records with high confidence. The authors further state that their attack can also be used against synthetically generated data simply by computing the underlying statistics from the generated microdata. Limitations of the work are that generally only some fraction of all records can be recovered with high confidence and that the confidence of resulting data records being part of the dataset D is just given via their order but not in absolute estimates.

2.9 Subsumption of the Privacy Debate and Technical Measures

In the last 15 years, a vivid discussion about disclosure risks which sensitive data about individuals can entail and the suitability of measures for reducing these risks has taken place in computer science and law and severely oppositional positions crystallized. In this section we summarize this discussion, integrate results covered in this chapter so far into the discussion and review some proposals for required policy changes in privacy regulation based on these results.

2.9.1 Discussion Summary

One of the first relevant articles is *Broken promises of privacy* by Ohm [Ohm09] who publicized the danger of re-identification attacks to the legal community. Ohm is influenced especially by the work of Narayanan and Shmatikov (see Section 2.8.4) and therefore considers the threat of re-identification attacks based on (even innocent-looking) background knowledge to be highly relevant. He proposes the passing of what he calls the *robust anonymization assumption* – the idea of robustly protecting individual's privacy with simple de-identification techniques. Instead, the *easy re-identification result* takes its place which doubts the abilities of simple release-and-forget anonymization due to the success of re-identification attacks. Based on these ideas, he argues against any PII-based regulation. Instead, he votes for rigorous scenario-dependent risk assessments regarding potential privacy harm from data disclosure based on factors like applied data protection techniques, data sensitivity and quantity, and the nature of potential adversaries. By weighing the identified risks against the potential data benefits one can determine if the data can be disseminated, if further risk-reducing measures have to be applied, or if the dissemination is not possible.

Yakowitz [Yak11] takes a mostly opposing stance to Ohm and focuses on the benefits of research based on public data. She underlines that the dissemination of anonymized data has influenced most of public policy debates, such as health insurance or census microdata. Often this data was collected for unrelated purposes. She concludes that policymakers cannot determine potential contributions of data in advance and "any rule that significantly impedes the release of research data imposes a social cost of uncertain magnitude" [Yak11]. Furthermore, she highlights the chances of crowdsourcing by general and unrestricted access to research data. She criticizes the computer science literature about re-identification risks for inaccurate assertions. Amongst others, she disagrees with the general assumption that each attribute can pose as an indirect identifier (QID). In her opinion, law only focuses on indirect identifiers that potentially are in the public domain. Other sources of information (and especially self-revealed information about extroverted individuals) should not determine the bounds of data dissemination. Another critique is directed against the idea that the public datasets generally have value for adversaries. While she admits that it is possible for adversaries to learn sensitive information, the comparison to risks existing independent of the data dissemination. Examples for these risks include self-reported information, commercially collected consumer data, or the risks imposed by insecure data processing in the presence of intruders. Finally, she does not agree with the assumption that, in principle, everyone is able to perform re-identification attacks due to the expertise required by adversaries. Due to these reasons, she votes for general and easy to apply rules and punishment for re-identification. The proposed rules include minimum subgroup sizes of five individuals, that is, k-anonymity for $k = 5$, and random sampling. We have seen in Sections 2.5.4 and 2.6.8 that these measures entail several weaknesses.

Wu [Wu13] accuses both of misinterpreting several aspects of the relevant computer science literature and sees their disagreement as based on varying understandings of the concepts of *privacy* and *utility*. According to Wu, Ohm underestimates the chances of even simple de-identification techniques (due to misunderstandings related to an impossibility result by Dwork [Dwo06]) and DP (see Section 2.7), while Yakowitz relies on the vulnerable concept of k-anonymity and improperly downplays the success of re-identification attacks. Rubinstein and Hartzog [RH16] provide similar arguments and furthermore mention that Ohm as well as Yakowitz limit their analyses almost exclusively to what Ohm refers to as release-and-forget anonymization, while disregarding further measures like restricted access or interactive mechanisms.

Another contribution is provided by Cavoukian and Castro [CC14]. They argue for the effectiveness of de-identification in minimizing re-identification risk while preserving data utility and make arguments similar to the ones of Yakowitz. But they also admit that effective de-identification is not a simple task and might not be suitable for all situations (for example, for high-dimensional data) and propose restricted access in the form of *data enclaves* as a solution in these cases. In principle, they advocate a risk-based approach which does not guarantee full privacy protection but reduces the risks the data imposes on individuals in the dataset:

> While it is not possible to guarantee that de-identification will work 100 per cent of the time, it remains an essential tool that will drastically reduce

> the risk of personal information being used or disclosed for unauthorized or malicious purposes. [CC14]

Narayanan and Felten [NF14] criticize several aspects of the work by Cavoukian and Castro. In their opinion, the work ignores adversaries with non-public background knowledge as well as targeted re-identification attacks. It assumes (like Yakowitz) that only highly specialized experts can perform these attacks, while in the opinion of Narayanan and Felten simple programming and statistics skills are sufficient. Further, they disagree with the implicit *penetrate-and-patch* approach present in the work by Cavoukian and Castro: The risk-based approach just considers attack types and background knowledge at a particular time. Future weaknesses in terms of re-identification successes cannot be prevented once data is disseminated. In summary, they view de-identification as promoting a false sense of security as it fails in theory as well as in practice.

Further publications include the one of Bambauer, Muralidhar, and Sarathy [BMS13] (which we have already covered in Section 2.7.7), several ones taking a more mediating stance [SS11; PTF16] just like Wu [Wu13] and Rubinstein and Hartzog [RH16], and others who propose formal solutions to legal questions [Nis+17; CN20; Alt+21] (see Section 2.9.3). Another example for the different positions is provided by an exchange in *Science* [Bar+15; MP15].

Rubinstein and Hartzog [RH16] summarize the discussion by identifying two opposing schools of thinking, they term *pragmatists* and *formalists*. They attribute the disagreements to the “very different histories, questions, methods, and objectives” [RH16] of the originating disciplines. The pragmatist position is determined by valueing “practical solutions for sharing useful data to advance the public good” [RH16]. Pragmatists focus on de-identification methods and assess the progress in re-identification attacks (see Section 2.8) as mostly academic. Formalists, on the other hand, judge simple de-identification techniques as providing a false sense of privacy due to their somewhat artificial adversary models. Additionally, formalists view examples of re-identification attacks as proof for the insufficient guarantees of simple de-identification techniques. As a consequence, they insist on quantifiable and mathematically provable privacy guarantees.

2.9.2 Technical Background for the Discussions

In the light of these discussions and to understand why they arise, we provide an overview of potential for misunderstandings as well as advantages and disadvantages of privacy techniques covered in this chapter and beyond it.

Section 2.1 has shown that the term (data) privacy comprises a variety of concepts. A violation of privacy can mean the association of a data record with an individual with 100 % certainty (*deterministic identity disclosure* in our introduced terminology), an increase in an adversary’s belief about a specific sensitive attribute value for an individual (*probabilistic positive attribute disclosure*), or the information that an individual is part of a dataset (*deterministic positive membership disclosure*). Therefore, even the concepts behind the discussion vary to a large extent. Wu [Wu13] elaborates on the different understandings of privacy, threats, and adversaries (see Section 2.1).

Similarly, also the understandings of terms used in the discussion and concepts behind these terms can diverge greatly (see Section 2.3). A central example is the definition of PII which serves as the most important concept forUS privacy laws. Since the definition of PII is directly related to identifiability, it comes at no surprise that the understandings differ a lot depending on the assessment of re-identification risks. From these varying understandings, it follows that also the interpretation of practical risks and the appraisal of past re-identification attacks (see Section 2.8) can vary to a large degree.

This holds true for the rating of the effectiveness of privacy techniques as well, as we have seen in the illustrated discussions. This is not astonishing, since they all involve advantages as well as disadvantages with respect to privacy and utility aspects (see also Section 2.2). First, we shortly summarize the privacy details of techniques covered in this section. Pseudonymization (see Section 2.4) allows for the linking multiple data records and re-identification of individuals, but provides no data privacy guarantees. Simple de-identification techniques (covered in Section 2.5), while giving a feeling of protection, generally do not allow for provable privacy guarantees. For this reason, most of the re-identification attacks covered in Section 2.8 are associated with de-identified datasets. Syntactic privacy models (see Section 2.6) introduce measurable guarantees with respect to specific disclosure risks, but do not come without problems. The classification of attributes depends on potential current and future background knowledge of adversaries, the necessary transformations of high-dimensional data can result in large utility losses and there is a number of known attacks on these models. Finally, semantic privacy models and DP in particular (see Section 2.7) provide provable and quantifiable privacy guarantees independent of adversarial background knowledge and capabilities, which compose nicely. But practical issues, such as the question of how to choose and split the privacy budget ε for multiple users and queries, can impede the wider use of DP. Furthermore, DP is not suitable for small datasets or the studying of outliers.

But the discussions also illustrate the necessity to consider aspects of utility (or benefit) as well. We have to recognize the necessity to balance privacy with the public good that data-based research can create [CT13], in other words, we have to weigh the disclosure risks for individuals with the social significance of the information. On the technical side there are some ways to measure the utility loss of specific privacy models (see Section 2.2). But there are also less technical and more practical considerations to take. First and foremost, carelessly applied privacy techniques can drastically reduce the utility of data or even render datasets unusable. Furthermore, privacy techniques can require a change in the way data users perform their tasks [DE13]. If a technique allows the processing of data with sufficient privacy guarantees and utility, but requires practitioners to radically change the way they perform their research, for example, by requiring other methods or preventing the use of established tools, this might prevent the research from taking place at all. Another aspect is that research often is based on exploratory data analysis [Agg+05] and specifics of datasets, such as data errors, unusual data distributions, or unexpected correlations, are easier to assess with direct data access [DE13]. Methods based on interactive DP prevent this access. Additionally and in comparison to the simplicity of the release-and-forget model, they require to provide a query interface, pay for server resources, and perform regular updates and audits [NS10]. On the other hand, for non-interactive data publishing it would be required to know in advance which information should be considered (most) socially useful and to choose the privacy protection measures accordingly to maximize the utility. But this is no practical assumption and instead, one aims at supporting multiple uses simultaneously [Wu13], which potentially reduces the possible utility. Other scenarios to consider are the ones in which researchers might want to be able to contact individuals. Reasons for this include, amongst others, re-feeding research results for a better treatment outcome, warning about drug side effects, or finding patients with unusual attributes relevant for research. Finally, the influence of privacy techniques can be hard to understand, as we have exemplarily presented for the case of DP in Section 2.7.7. A lack of comprehensibility can lead to disregarding research results, even if they might actually be valid.

2.9.3 Proposals for Policy Changes

As we have seen in the previous section, there is no silver bullet in data privacy. The advantages in re-identification research have resulted in the need for new regulatory paradigms [NS19]. Narayanan and Felten [NF14] see three alternative approaches for data custodians:

- Using de-identification measures and hope for the best.
- Switching to provable guarantees like the ones DP provides and handle the potential losses in utility and convenience.
- Relying on legal agreements that limit the dissemination and use of sensitive data.

None of these solutions (and no combination of them) is viewed as fully satisfactory or best suited for all situations and “policy makers must confront hard choices” [NF14]. Based on this insight, several scholars (from law as well as computer science) vote for multi-layered approaches using suitable technical and organizational measures based on thorough risk assessments on a case-by-case basis.

Cormode [Cor11] argues for nuanced threat models, taking into account potential adversaries, data recipients, perceived threats, and consequences of successful attacks. Depending on this modeling, the suitable measures can include de-identification, syntactic privacy models, semantic privacy models, or refrain from release completely. Garfinkel [Gar14] differentiates release models to reduce the risk of re-identification. In addition to the release-and-forget model, these include the use of data use agreements (DUAs), and the enclave model employing interactive queries. Dwork, Kohli, and Mulligan [DKM19] mentions the possibility of legally constrained data access, use, and sharing, especially for small datasets for which DP is not a suitable technical measure.

According to Narayanan, Huey, and Felten [NHF15] there is no one-size-fits-all solution for data privacy since each dataset entails a distinct risk-benefit tradeoff. One needs to balance privacy threats and the expected damage from sensitive information leakage with benefits from wider data access and improved analysis. The main problem is that one in this case has to weigh uncertain risks against uncertain benefits. Due to the unknowable risks of simple de-identification techniques, the authors argue against the default policy of allowing unrestricted public data release in many privacy regulations based on the PII concept. Instead, they favor two alternatives: using provable privacy methods like DP or restricting the access to datasets to a narrow audience. They provide several aspects to consider when choosing the adequate scope for the dataset release, amongst others:

- Is it possible to use techniques with provable privacy guarantees?
- Can the data holder provide an interactive query service instead of data release?
- Are there aggregate statistics providing similar benefits to microdata release?
- Are there subsets of the general public, such as researchers, most likely to achieve the benefits?
- Is it possible to provide different forms of the dataset with distinct privacy risks and utility according to the recipient's needs and trustworthiness?
- Can the recipient be required to sign data use agreements?

These and similar aspects can be used to determine an appropriate release scope to prevent the risks of unrestricted data dissemination.

Another proposal for necessary policy changes is provided by Rubinstein and Hartzog [RH16], who focus on the process of minimizing privacy risks. They acknowledge that there is no perfect anonymity as well as that de-identification is severely limited and vote for using the full spectrum of (technical and legal) privacy measures, including differential privacy, tiered access, DUAs, and even educational efforts. One central aspect of their proposal is the evolution of current outcome-based data privacy regulations to something similar to the process-based data security regulation, especially with respect to three key ideas:

- The regulation should focus the process of protecting individuals by using adequate privacy techniques in combination with access and use restrictions.
- The necessary protection level depends on contextual factors, including potential harms resulting from data use, adversarial motivation and abilities, desired data utility, and the possibility to use further risk minimization controls.
- There is no perfect security, so the goal should be to reduce the privacy risks respectively increase the costs for re-identification to acceptable levels.

Due to the contextual dependency on multiple factors, there is no one-size-fits-all standard or the possibility of detailed checklists, which do not get outdated fast. For this reason, the authors suggest (similar to data security regulations) to rely on industry standards to assess the reasonability of processes – even though there exists no globally accepted standard at the moment. Finally, they reason that release-and-forget anonymization is rarely (if ever) an acceptable strategy, since risk assessment is difficult when one loses control over the shared data.

Another recent line of research [Nis+17; CN20; Alt+21] tries to deduce formal requirements from law. The goal is not a formalization of regulations but to provide support for practitioners and regulators via formal tools, in other words, a "hybrid legal-technical approach to the evaluation of technical measures to render information anonymous" [Alt+21]. This can help to reduce uncertainty or even contradictory recommendations regarding interpretation of regulations. Otherwise this uncertainty can often lead to increased disclosure risks due heuristic processes being implemented [Nis+17].

Nissim et al. [Nis+17] provide an approach to bridge the gap between privacy requirements of FERPA and DP. They extract an adversarial privacy game from the law, its history, and official guidance. This game can be used to test whether computations meet the privacy requirements of FERPA. Another approach is introduced by Cohen and Nissim [CN20] and complemented by Altman et al. [Alt+21]. They consider the *singling-out* principle, which is a fundamental principle with respect to anonymization in the GDPR (see Sections 2.3.1 and 2.3.2), by means of so-called predicate singling out (PSO) security. The basic idea is to consider predicates $p : X \rightarrow \{0, 1\}$ for individuals X in a dataset D. A mechanism M is PSO secure, if an adversary is not able to find a predicate p which singles out a unique individual in $M(D)$, that is, $p(X_i) = 1$ and $p(X_j) = 0$ for all $i \neq j$. The authors show that k-anonymity and further syntactic privacy models do guarantee PSO security, DP, on the other hand, might do. But the additionally argue that PSO security is a necessary but insufficient property for secure anonymization due to further threats, such as membership disclosure or attribute disclosure.

After this comprehensive overview of the field of privacy-preserving data publishing, in the following chapters we investigate the application of specific techniques in distributed environments. We begin by presenting an approach for the generation of pseudonyms for data records collected at various data sources in Chapter 3.

3 Distributed Global Pseudonyms

Medical data from patients is collected in a variety of different places, for example by family doctors, in treating hospitals, in rehabilitation facilities, or by insurance companies. In some situations, the integration of data from these data sources is essential for medical researchers to conduct comprehensive studies and gain insights into complex medical conditions and treatment outcomes. For example, to assess the long-term success of a treatment it is not sufficient to only look at data collected during the patient's stay at the hospital. One has to link this data with the data collected during rehabilitation (and potentially even in later years) to evaluate the success.

However, leveraging these disparate data sources for research purposes imposes technical and regulatory challenges. Protecting patient privacy is of utmost importance when working with medical data and the combination of multiple data records related to one patient can drastically increase the privacy risks for the patient. In addition, there may be legal restrictions that prevent the data from being combined directly. Pseudonymizing medical data is a common practice to reduce privacy risks while still enabling data analysis. Pseudonymization involves replacing identifying information, such as names or social security numbers, with unique identifiers (see Section 2.4). This allows researchers to link data from different sources without learning the patient's identity. While the pseudonymization of local data is relatively simple, the pseudonymization of distributed data in a way that a patient gets assigned the same pseudonym at all data sources (we will refer to this as *global pseudonym consistency*) requires more thought – especially when no single party should be able to access all identity-pseudonym relationships.

In this chapter, we present a solution to the problem of distributed pseudonymization. For this purpose we utilize SE – a technique which enables protected search queries over encrypted data. Our solution improves on related work in that individual data sources can be included and removed from the pseudonymization process easily, while supporting so-called fuzzy search capabilities as well as ways to limit the linkability of data records.

Our main contributions are the following:

- We provide a scheme for distributed pseudonymization based on SE.
- We extend the basic scheme with fuzzy search capabilities to allow the linkage of patient's data even in the presence of different spellings or typos.
- We provide ways to limit the linkability of data records with respect to time or budget restrictions.

In Section 3.1 we provide background information about SE and the specific SE scheme we utilize in this chapter. Section 3.2 explains specifics of the scenario and proposes several properties which schemes for globally consistent pseudonyms should entail. In Section 3.3 we introduce our SE-based scheme for single identifying attributes as well as extensions for multiple attributes and fuzzy search. Furthermore, we cover options for limiting the linkability in the scheme. We present our adversary model for the scheme in Section 3.4. In Section 3.5 we describe the scheme implementation and evaluate its performance. Section 3.6 compares our scheme to related work based on the properties given in Section 3.2. We conclude the chapter in Section 3.7.

This chapter builds up on work by Zimmer et al. [Zim+20] which in turn is based on parts of my master thesis [Pet18]. The scheme presented in Section 3.3 was initially implemented by Herbst [Her23] in his bachelor's thesis and he also developed the approach for fuzzy search capabilities (see Section 3.3.3).

3.1 Background

In this section we provide the technical background for the SE protocol we employ in our solution to the problem of globally consistent pseudonyms. Section 3.1.1 provides an introduction to SE. Section 3.1.2 shortly describes *bilinear pairings* as a prerequisite for the specific employed SE protocol we detail in Section 3.1.3.

3.1.1 Searchable Encryption

SE describes a family of schemes invented for the scenario of encrypted remote storage. One or multiple users store their data in encrypted form on a server so that this server has no access to the plaintexts. SE techniques enable one or multiple users to search this data remotely.

The following general model of SE provided by [Bös+14] formalizes this setting. An overview of this model is provided in Figure 3.1. Most[1] of the SE schemes are based on *search indices* a user has to provide alongside with the encrypted data. For a collection of documents $D = (D_1, \ldots, D_n)$ a list of searchable keywords $W = (w_1, \ldots, w_m)$ is extracted. The SE algorithm `BuildIndex` provided with a user key k_S, the documents D, and the keywords W computes the search index I – a data structure which enables the server to perform searches without access to plaintext documents or search keywords. Another SE algorithm `Enc` is used with another user key k_E to encrypt the documents D, resulting in ciphertexts C. The search index I and the encrypted documents C are then stored at the server. When a search is to be performed, the user sends a specific search request T (often referred to as *trapdoor*) computed by the SE algorithm `Trapdoor` using the key k_S and a search keyword w (or a search predicate based on the keyword) to the server. The server can query the search index I based on this trapdoor T using the SE algorithm `Search`, find the matching documents C_w and return them to the user. These received encrypted documents can be decrypted using the SE algorithm `Dec` with the user key k_S resulting in documents D_w. All this is achieved in a way that does not allow the server to learn the content of the documents D_w or the search term w.

One can differentiate between four categories of SE schemes depending on the potential number of writers and readers in the scheme [Bös+14]. The most researched categories

[1] One example of an exception is the first ever searchable symmetric encryption (SSE) scheme introduced by Song, Wagner, and Perrig [SWP00]. They employ a specifically crafted symmetric encryption scheme which allows to perform search operations on ciphertexts directly.

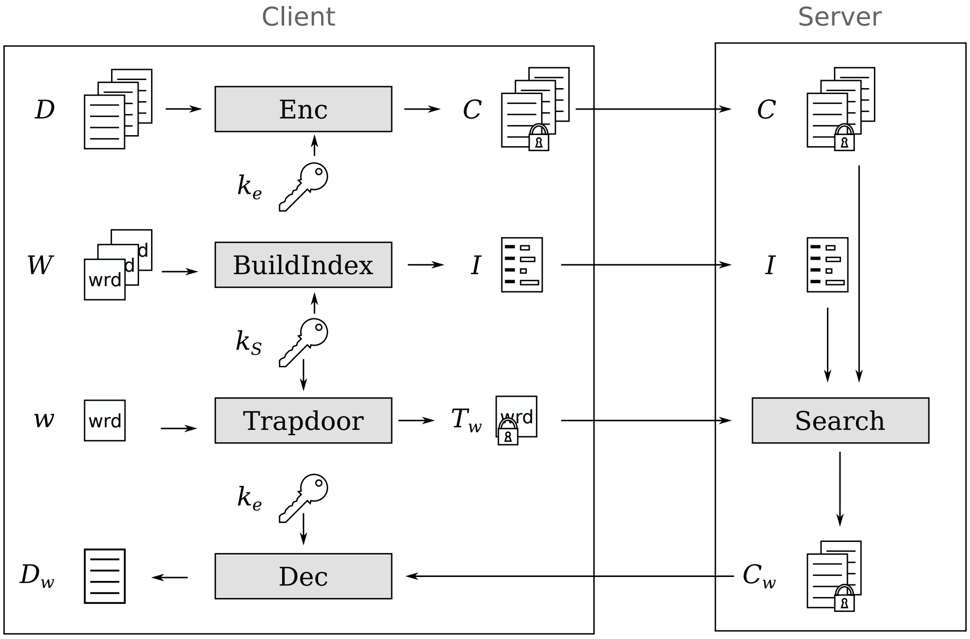


Figure 3.1: General model of index-based SE schemes.

are the *single-writer/single-reader* and the *multi-writer/single-reader* categories. *Single-writer/single-reader* schemes are generally referred to as SSE schemes. These schemes focus on the scenario of personal data to be stored on a not-fully trusted remote server in a searchable manner. Respective keys used for searching and encryption are kept by the data holder. Pioneering work in the field of SSE has been performed by Song, Wagner, and Perrig [SWP00], Goh [Goh03], Chang and Mitzenmacher [CM05], and Curtmola et al. [Cur+06]. *Multi-writer/single-reader* schemes are generally known as public key encryption with keyword search (PEKS)[2]. The main usage scenario for these schemes is searchable storage of emails and alike. Senders use the public key of the receiver and together with the encrypted document they store a search index based on keywords of the respective document. Only the receiver in possession of the private key is able to create trapdoors to search through all documents. The first PEKS scheme was introduced by Boneh et al. [Bon+04], who also coined the term PEKS. Other research extends the field with schemes allowing multiple *readers* to perform searches on encrypted data. Examples for these schemes can be found in the survey by Bösch et al. [Bös+14]. Our technical solution to the problem of globally consistent pseudonyms is based on a multi-writer/multi-reader scheme introduced by Bao et al. [Bao+08] detailed in Section 3.1.3.

Apart from the supported reader/writer setting, schemes can be differentiated by further properties [Poh+17]. They can support various *query functionalities*, ranging from simple single keyword searches to conjunctive multi-keyword queries to fuzzy or range queries. Some schemes (*dynamic* schemes) are able to update the documents and respective indices, that is, add or remove individual documents, while others (*static* schemes) are not. Further properties include the verifiability of operations, performance of scheme algorithms, size of generated indices, and achieved security model[3]. For

[2] Some publications refer to PEKS as *asymmetric searchable encryption* [ATR14].

[3] There are different security definitions for SE such as *semantic security against adaptive chosen keyword attacks* (IND1-CKA) [Goh03]. Further definitions were provided by Chang and Mitzenmacher [CM05],

the interested reader there are several surveys [Bös+14; WWC16; Poh+17; ZXL18], which can be contacted for further information about SE, specific schemes, and their properties.

3.1.2 Bilinear Pairings

The following definition is given by Menezes [Men09].

Definition 3.1.1. For a prime p, a group G_1 of order p with generator P and identity ∞ (written in *additive* notation), and a second group G_T of order p with identity 1 (written in *multiplicative* notation), a *bilinear pairing* $\hat{e} : G_1 \times G_1 \to G_T$ is a map satisfying the following three conditions:

1. $\forall R, S, T \in G_1 : \hat{e}(R + S, T) = \hat{e}(R, T) \cdot \hat{e}(S, T)$ and $\hat{e}(R, S + T) = \hat{e}(R, S) \cdot \hat{e}(R, T)$ (bilinearity).
2. $\hat{e}(P, P) \neq 1$ (non-degeneracy).
3. $\hat{e}$ can be efficiently computed (computability).

The security of pairing-based protocols is commonly based on the hardness of the bilinear Diffie-Hellman problem (BDHP): For a bilinear pairing $\hat{e}$ on groups G_1, G_T and given values P, aP, bP, cP, compute $\hat{e}(P, P)^{abc}$. The SE protocol we use in this chapter (see Section 3.1.3) relies on the fact, that the computational Diffie-Hellman problem (CDHP) in G_1 (given P, aP, bP, compute abP) is still hard, but the decisional Diffie-Hellman problem (DDHP) in G_1 (given P, aP, bP, cP, is $cP = abP$?) is efficiently solvable with the aid of $\hat{e}$. Further details are provided by Menezes [Men09].

3.1.3 A Multi-Reader/Multi-Writer Searchable Encryption Scheme

Bao et al. [Bao+08] provide a multi-reader/multi-writer SE scheme. It allows a set of users to write records to an encrypted database and to perform searches over all records, that is, records created by a user themself as well as records created by other users. For this purpose the scheme relies on two components: A database server $Serv$, which stores encrypted records and performs the search, and a user manager UM, which is responsible for user enrollment and revocation.

The multi-user scheme consists of several algorithms[4]. `Setup()` for setting up systems parameters and generating key material, `Enroll()` for adding new users to the system and providing them with required keys, `GenIndex()` for generating indices required for searchable encrypted records, `Write()` for storing encrypted records and their respective indices, `ConstructQ()` for constructing search queries, `Search()` for performing a search query on the database, and `Revoke()` for removing search capability from users.

Curtmola et al. [Cur+06], and Boneh et al. [Bon+04]. Overviews are provided by Bösch et al. [Bös+14] (Section 2.3) as well as Poh et al. [Poh+17] (Section 4).

[4] Bao et al. speak of algorithms here, even though algorithms `GenIndex()` and `Write()` would be better described as interactive protocol steps. For the sake of simplicity, however, we stick to the terminology introduced by Bao et al.

The scheme is based on bilinear pairings, for which we use the notation introduced in Section 3.1.2. Let p be a prime, G_1 a group of order p with generator P and identity ∞ (written in *additive* notation), G_T a second group of order p with identity 1 (written in *multiplicative* notation), and $\hat{e} : G_1 \times G_1 \rightarrow G_T$ a *bilinear pairing*. Let $h : G_T \rightarrow \mathcal{K}$ denote a cryptographic hash function mapping elements in G_T to the key space $\mathcal{K}$ of a symmetric encryption scheme and $h_S : \mathcal{S} \times \mathcal{W} \rightarrow G_1$ a keyed hash function mapping a keyword $w \in \mathcal{W}$ to an element of G_1 under a seed $s \in \mathcal{S}$. Further, let $E : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}$ and $D : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}$ denote the encryption and decryption operations of a symmetric encryption scheme with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$. For cleaner presentation, we use the following additional notations: $h_s(w) = h_S(s, w)$, $E_k(m) = E(m, k)$, and $D_k(c) = D(c, k)$. In the following, we provide details for each algorithm as introduced by Bao et al. [Bao+08].

- `Setup`(1^κ): This algorithm is executed by the user manager UM to set up the system. For a given security parameter κ, it picks public parameters G_1, G_T, and $\hat{e}$ and creates the user manager key k_{UM} from $\mathbb{Z}_p*$, the symmetric encryption key $e \in \mathcal{K}$, and the seed $s \in \mathcal{S}$ for h_s randomly.

- `Enroll`(k_{UM}, u): This algorithm is executed by the user manager UM to enroll a new user given by their identity u. The user manager UM includes u in the list of authorized users $\mathcal{U}_A$, selects a random $x_u \in \mathbb{Z}_p*$ and computes a pair of related keys specific for user u: the query key $qk_u = (x_u, s)$ and the complementary key $ComK_u = (k_{UM} - x_u)P$. User u receives qk_u and e. The database server $Serv$ receives $ComK_u$ and stores it together with the user identity u in a list of authorized users.

- `GenIndex`($qk_u, w; ComK_u$): This algorithm is executed interactively by user u and the database server $Serv$ to generate an index for the keyword w. User u selects a random blinding element r_w from $\mathbb{Z}_p*$ and sends a *generate index request* $(u, r_w \cdot h_s(w))$ to $Serv$. The database server $Serv$ returns $e_w = \hat{e}(r_w \cdot h_s(w), ComK_u)$ to u. The user then computes $k = h((e_w)^{\frac{x_u}{r_w}})$ and the final index $I_w = \langle r, E_k(r) \rangle$ for a random $r \in \mathcal{M}$.

- `Write`($qk_u, e, d_i; ComK_u$): This algorithm is executed interactively by user u and the database server $Serv$ to write an encrypted record to the database D'. For a document d_i, the respective keyword $d_i.w$ and the index $I_{d_i.w}$ (computed via `GenIndex()`), user u sends $d_i' = \langle E_e(d_i), I_{d_i.w} \rangle$ to $Serv$. The database server $Serv$ appends this tuple to the database D'.

- `ConstructQ`(qk_u, w): This algorithm is executed by user u to construct a query for a keyword w. For this the user, in possession of the query key $qk_u = (x_u, s)$, computes the query $q_u(w) = x_u \cdot h_s(w)$.

- `Search`($q_u(w), ComK_u, D'$): This algorithm is executed by the database server $Serv$ to perform the search over all encrypted database records D'. For a query $(u, q_u(w))$, the database server $Serv$ looks for $ComK_u$ and computes $k' = h(\hat{e}(q_u(w), ComK_u))$. It then adds all encrypted records $d_i' = \langle E_e(d_i), I_{d_i.w} \rangle$ with $I_{d_i.w} = \langle r, E_k(r) \rangle$ to the query result set, for which $r \overset{?}{=} D_{k'}(E_k(r))$. Finally, it outputs the result set $a_q = \{E(d_i) \in D' | d_i \in D, d_i.w = w\}$.

The search algorithm is correct, in other words, the search result set only contains records d_i with $d_i.w = w$, if $k = k'$, meaning that $Serv$ can successfully decrypt $E_k(r)$. This is quite simple to show. For a keyword w and the respective query $q_u(w) = x_u \cdot h_s(w)$ we obtain:

$$\begin{aligned} k' &= h(\hat{e}(q_u(w), ComK_u)) \\ &= h(\hat{e}(x_u \cdot h_s(w), (x - x_u)P)) \\ &= h(\hat{e}(h_s(w), P)^{\frac{x \cdot x_u}{x_u}}) \\ &= h(\hat{e}(h_s(w), P)^{x}) \end{aligned}$$

Comparing this to the key k computed by the client during index generation results in the same key (for the same keyword w):

$$\begin{aligned} k &= h((e_w)^{\frac{x_u}{r_w}}) \\ &= h(\hat{e}(r_w \cdot h_s(w), ComK_u)^{\frac{x_u}{r_w}} \\ &= h(\hat{e}(r_w \cdot h_s(w), (x - x_u)P)^{\frac{x_u}{r_w}} \\ &= h(\hat{e}(h_s(w), P)^{r_w \frac{x}{x_u} \frac{x_u}{r_w}} \\ &= h(\hat{e}(h_s(w), P)^{x}) \end{aligned}$$

- `Revoke`(u): This algorithm is executed by the user manager UM to remove the search capabilities for user u. It removes u from the list of authorized users $\mathcal{U}_A$ and instructs $Serv$ to delete the entry $(u, ComK_u)$ from the respective list.

Bao et al. [Bao+08] provide proofs for three security properties of this scheme in the honest-but-curious attacker model [PMB14]:

- **Query privacy**: The database server $Serv$ does not learn any information, apart from observable database access patterns[5], from a query, especially not the keyword w. This property does not hold under collusion of users and servers.

- **Query unforgeability**: Neither other users nor the database server $Serv$ are able to craft valid search queries for user u without access to their search key qk_u.

- **Revocability**: Revoked users are no longer able to perform searches.

3.2 Scenario and Properties

In this section we provide the general scenario for globally consistent pseudonymization. Additionally, we present several important properties for technical solutions achieving this kind of pseudonymization.

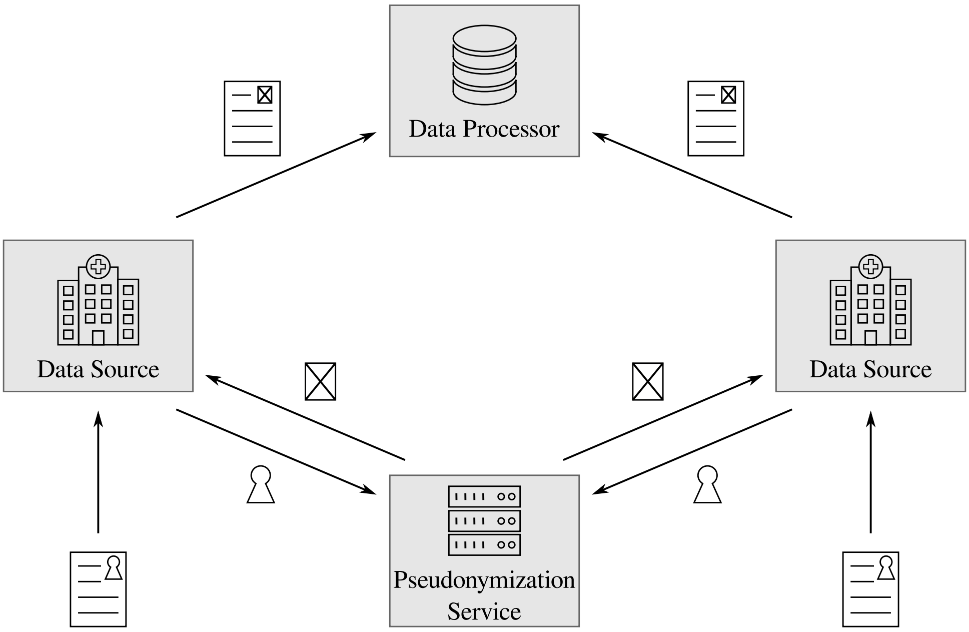


Figure 3.2: Scenario of globally consistent pseudonymization

3.2.1 Scenario

The scenario we focus in this chapter is based on a number of distinct *data sources* in which data records related to individuals arrive continuously. A data record contains *identifying attributes* which have to be pseudonymized as well as additional attributes required to be processed for further purposes, such as data-based research. The key problem is that individuals potentially provide data records at multiple data sources. The goal of this chapter is to pseudonymize data records in a way that data records relating to the same individual are assigned the same pseudonym – regardless of the data source. Following Zimmer et al. [Zim+20] and Lehmann [Leh19] here, we call these pseudonyms *globally consistent*.

To achieve this global consistency, there are two basic ideas [Zim+20]:

- Pseudonyms are derived from identifying data alone in a deterministic fashion without any communication with other parties, for example, by employing a hash function (see Section 2.4.4). Zimmer et al. [Zim+20] refer to this as *local deterministic pseudonymization*.

- Pseudonyms are created fully random and some sort of global lookup table maps identifying attributes to these pseudonyms. Zimmer et al. [Zim+20] call these *truly random pseudonyms*.

However, local deterministic pseudonymization opens up the possibility to perform dictionary attacks due to typically small attribute domains (see Sections 2.4.3 and 2.4.4). For this reason, we focus on truly random pseudonyms. Another party, we call *pseudonymization service*, is responsible for creating random pseudonyms or providing existing pseudonyms for the data records based on the identifying information.

[5] As stated by Curtmola et al. [Cur+06], any SE scheme reveals certain information, for example, two records sharing the same keyword, to the performing server.

Translating this scenario in the medical domain as an example results in the following description depicted in Figure 3.2. Patient data records, consisting of identifying information and medical information[6], are collected at medical centers such as hospitals, insurance companies, general practitioners, and other medical facilities. These medical centers contact the pseudonymization service to obtain a pseudonym based on the identifying information and replace this information in the data record with the pseudonym. Afterwards the pseudonymized data record is sent to the data processor. This party can study medical research questions after correlating data records about the same patient via the globally consistent pseudonym. Incorporating medical data from multiple data sources allows for the answering of more comprehensive research questions in comparison to research based on data from only a single data source.

3.2.2 Properties

There are several relevant properties a system which offers globally consistent pseudonyms might entail. Zimmer et al. [Zim+20] proposed several properties which focus on preventing participants of the pseudonymization process from learning unnecessary information (*data minimization*). We extend these properties with further ones which play an important role in practical scenarios.

- **Attribute Confidentiality**[7] [Zim+20]: During the pseudonymization process no party apart from the data source itself learns information about the identifying attributes the pseudonym is created for. This property does not cover information which another data source obtains by processing the same identifying attribute in another pseudonymization process.

- **Re-Use Indistinguishability** [Zim+20]: During the pseudonymization process a data source obtains no information about how often a pseudonym has already been provided to other data sources. This includes not being able to distinguish between already existing and newly created pseudonyms.

- **Matching Pseudonym Unobservability** [Zim+20]: The pseudonymization service does not learn the resulting pseudonym for a request during the pseudonymization process.

- **Limited Linkability** [Zim+20]: Data records relating to the same individual receive the same pseudonym only for some given validity period. This can be represented by a time period, a maximum number of usages (budget), or some other property.

- **Fuzzy Search Capability**: The pseudonymization process allows to match similar, but not equal, identifying attribute values relating to the same individual to the same pseudonym. This property is relevant in practice where small differences like typos or other inhomogeneities in attribute values can occur [LBÜ15].

[6] In the German medical domain these concepts are often referred to as *IDAT* and *MDAT*. See Section 6.2.1 for a short discussion of this distinction.

[7] Zimmer et al. [Zim+20] use the term *deposit confidentiality* here and also introduce another party called *depositor* as the pseudonymizing component at the data source. But *attribute* confidentiality is a better name for this property, since it directly indicates the type of the stored deposit.

- **Multiple Data Sources**: The number of data sources participating in the pseudonymization process supported by the system is not constrained.
- **Manageable Data Sources**: Individual data sources can be enrolled in as well as revoked from the system. This is achieved without requiring any changes, such as updated cryptographic keys, for other data sources, which prevent already provided pseudonyms from being reused.

3.3 Globally Consistent Pseudonyms via Searchable Encryption

Prior work [Zim+20] looked at the problem of globally consistent pseudonymization, but left out the important question of key management. The authors assume a shared symmetric key, but provide no further details: "privacy-enhanced event pseudonymisation with limited linkability (PEEPLL) utilises HMACs by equipping all data sources Depositors with a shared secret k not known to the PVault[8]" [Zim+20]. This leaves open several substantial questions with respect to key management, amongst other things:

- How are new data sources enrolled in the system and how do they receive the required key material?
- How can the system revoke single data sources and prevent them from querying pseudonyms?
- How can the system handle disclosed or lost keys?

Two simple approaches are the collaboration of all data sources to collectively generate key material or the utilization of a TTP, which generates and distributes the key material. The collaboration approach, however, requires communication between all data sources which might not be feasible in all scenarios[9] and can impede a large communication overhead. Relying on a TTP prevents these disadvantages, if there exists a trustworthy party in a given scenario. But both approaches provide no simple answers to the questions of how to revoke users and how to handle disclosed key material. Generating new key material in these cases (collaboratively or through the TTP) would render already created pseudonyms meaningless, but sticking to the old key material would still allow adversaries in possession of these keys to query pseudonyms.

In this section, we provide a way to handle the problem of enrolling and revoking data sources by employing the SE scheme by Bao et al. introduced in Section 3.1.3. We start with a simple variant, in which each pseudonym holder is uniquely identified by a single identifying attribute. Afterwards we look at a more complex scenario, where a pseudonym holder can be identified by a combination of multiple identifying attributes, while these combinations not necessarily uniquely identify a pseudonym holder. Based on this extension we look at a variant, which includes fuzzy search capabilities for determining pseudonyms. Finally, we present options for limiting the linkability of data records in given schemes.

[8] *PVault* is their description for our pseudonymization service.

[9] One counterexample is the practical reality in sensitive areas of hospital networks where allowed communication connections have to be created manually for each communication partner, for example via firewall rules.

3.3.1 Solving the Key Distribution Problem

Our basic idea is to employ the keyword search functionality of the SE scheme for the detection of already existing pseudonyms by treating identifying attributes as keywords. Since in our scenario all data sources should be able to create new pseudonyms as well as query existing pseudonyms, we need to use a multi-reader/multi-writer SE scheme (see Section 3.1.1). One of the few schemes in this field is the one of Bao et al. [Bao+08] (see Section 3.1.3).

For the scenario of creating globally consistent pseudonyms for multiple data sources, we can adapt the scheme quite naturally. The user u in the SE scheme acts as a data source and the database server $Serv$ plays the role of the pseudonymization service. The user manager UM is introduced as a new party to our scenario, but participates only in enrolling or revoking users, not in querying or creating pseudonyms. The processing unit is responsible for the processing of pseudonymized data records and does not participate in the pseudonymization process. In this first variant, a data record is pseudonymized based on a single unique identifier id.

In the beginning, the system is set up by the `Setup()` algorithm, which defines public cryptographic parameters and creates required key material. Each data source u is then enrolled via the `Enroll()` algorithm, which provides them with the required key material qk_u. In comparison to the full scheme of Bao et al., for this variant we do not require the symmetric encryption key e, because it is not necessary to store the encrypted records themselves on the database server.

When a data source receives a new data record to pseudonymize, they extract the identifying attribute value id and perform a search using the value as keyword. They compute a search query using the `ConstructQ()` algorithm and send this query to the pseudonymization service. The pseudonymization service uses this query to perform the `Search()` algorithm and returns the query result set. This set consists of a potential matching index and the respective pseudonym P_{id}. We expect a single result when a pseudonym for the identifier has already been created and an empty result set otherwise.

If the result set is empty, the pseudonym for id has not been created yet. In this case, the data source u and the pseudonymization service perform the `GenIndex()` algorithm interactively for the keyword id, resulting in the index I_{id}. Afterwards, data source u sends this index to the pseudonymization service, who performs the `Write()` algorithm. In comparison to the full scheme of Bao et al., for this variant the data source does not send the symmetrically encrypted record, but just the index, because this index is enough to determine if the value id has already been assigned to a pseudonym. The pseudonymization service receives the index, generates a random pseudonym P_{id}, stores the resulting tuple $\langle P_{id}, I_{id} \rangle$ to the database D', and returns P_{id} to the data source.

When a new data source u_N is to be enrolled, this can be done just like during the initial setup phase via the `Enroll()` algorithm performed by UM. This process has no influence on enrolled data sources or existing pseudonyms. After receiving their key material, u_N can query all pseudonyms – including pseudonyms created before the enrollment of u_N.

Revoking a data source can be done by UM by performing the `Revoke()` algorithm. Afterwards, the revoked data source cannot query the pseudonymization service or

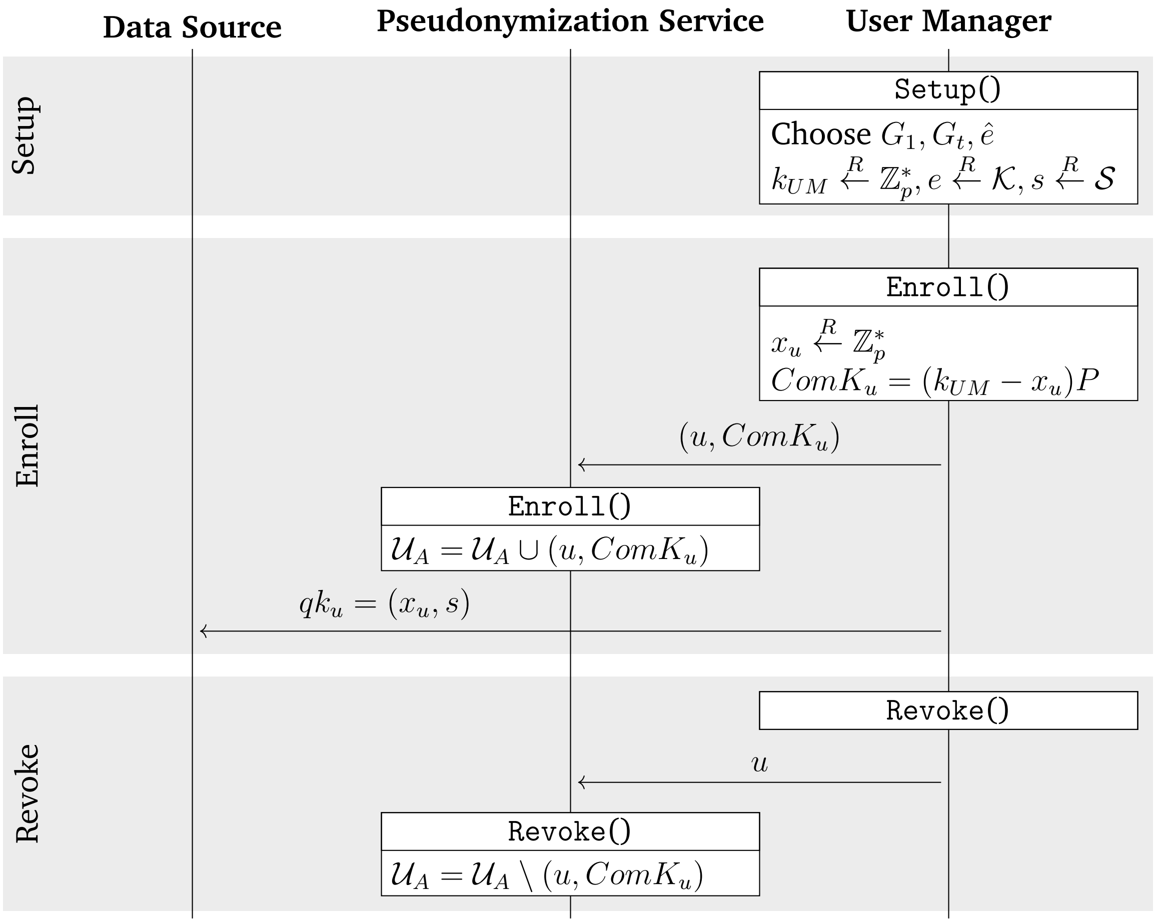

Data Source
Pseudonymization Service
User Manager
Setup
Enroll
Revoke
Setup()
Choose $G_1, G_t, \hat{e}$
$k_{UM} \xleftarrow{R} \mathbb{Z}_p^*, e \xleftarrow{R} \mathcal{K}, s \xleftarrow{R} \mathcal{S}$
Enroll()
$x_u \xleftarrow{R} \mathbb{Z}_p^*$
$ComK_u = (k_{UM} - x_u)P$
$(u, ComK_u)$
Enroll()
$\mathcal{U}_A = \mathcal{U}_A \cup (u, ComK_u)$
$qk_u = (x_u, s)$
Revoke()
u
Revoke()
$\mathcal{U}_A = \mathcal{U}_A \setminus (u, ComK_u)$

Figure 3.3: The initial setup, enroll and revoke operations in our protocol.

generate indices for creating pseudonyms anymore. Just like in the case of enrolling new data sources, this process has no influence on enrolled data sources or existing pseudonyms.

The full scheme is depicted in Figure 3.3 and Figure 3.4.

3.3.2 Extending the Scheme with Multi-Keyword Search

In the last section, we have only covered the case of one unique identifying attribute id being used for determining the pseudonym. But in reality there can occur scenarios, in which individuals are identified only by a combination of attributes, such as first name, last name, and date of birth. In these scenarios it might even be the case, that this combination of identifiers refers to multiple individuals. Another possibility would be, that individuals are identified uniquely by several distinct attributes, such as ID card number and health insurance ID. Therefore, in this section we extend the introduced scheme to multi-keyword search.

For multiple keywords, a data source has to provide a list of indices for a pseudonym for the `Write()` operation, where each of these indices was generated by the `GenIndex()` algorithm as described in the single-keyword case. For performance reasons, we can batch all index requests in a single communication round.

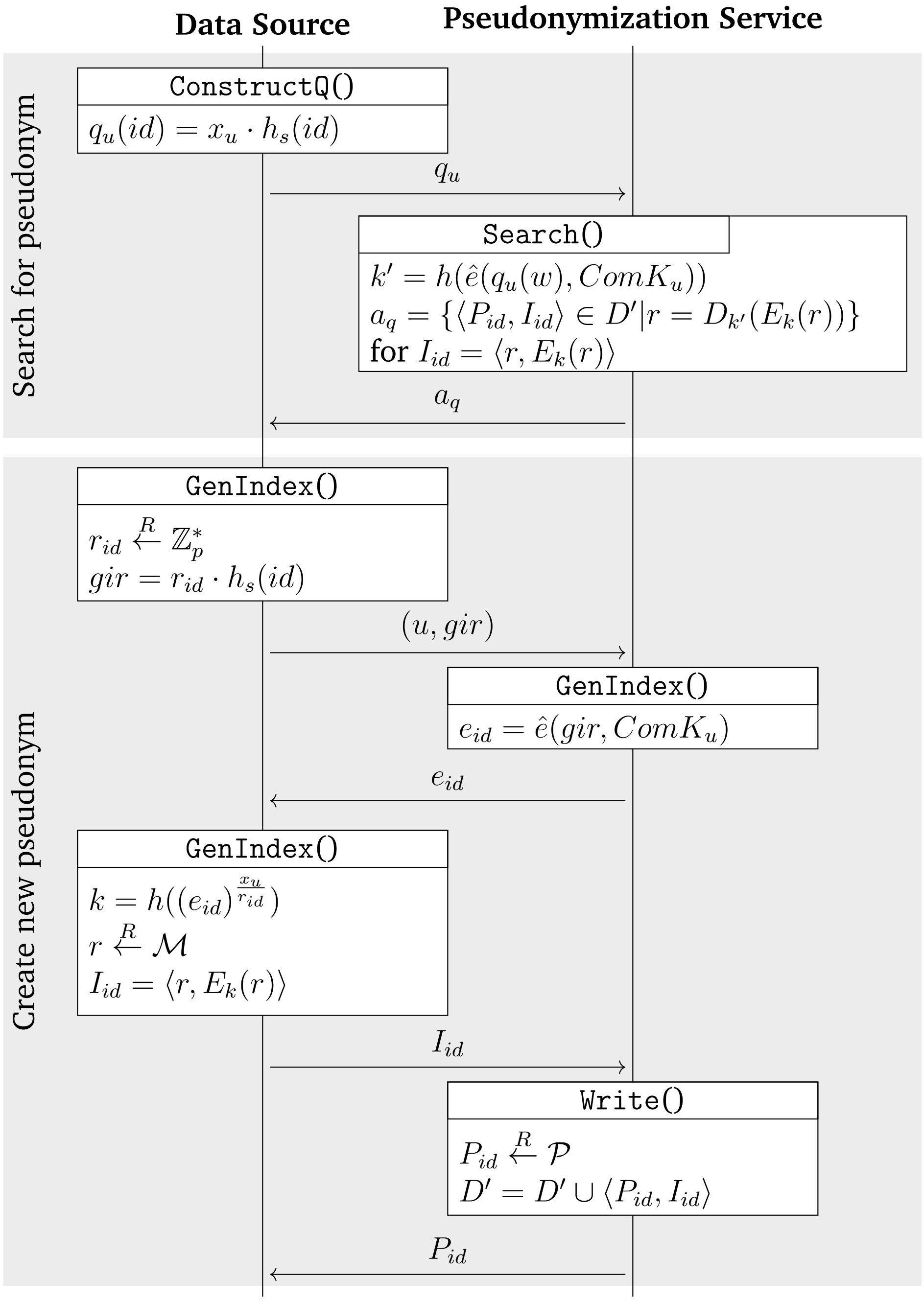

Data Source
Pseudonymization Service
Search for pseudonym
ConstructQ()
$q_u(id) = x_u \cdot h_s(id)$
q_u
Search()
$k' = h(\hat{e}(q_u(w), ComK_u))$
$a_q = \{\langle P_{id}, I_{id}\rangle \in D' | r = D_{k'}(E_k(r))\}$
for $I_{id} = \langle r, E_k(r)\rangle$
a_q
Create new pseudonym
GenIndex()
$r_{id} \overset{R}{\leftarrow} \mathbb{Z}_p^*$
$gir = r_{id} \cdot h_s(id)$
(u, gir)
GenIndex()
$e_{id} = \hat{e}(gir, ComK_u)$
e_{id}
GenIndex()
$k = h((e_{id})^{\frac{x_u}{r_{id}}})$
$r \overset{R}{\leftarrow} \mathcal{M}$
$I_{id} = \langle r, E_k(r)\rangle$
I_{id}
Write()
$P_{id} \overset{R}{\leftarrow} \mathcal{P}$
$D' = D' \cup \langle P_{id}, I_{id}\rangle$
P_{id}

Figure 3.4: Searching for and creating pseudonym operations in our protocol.

There are two possible ways to introduce multiple keywords in the database: as a simple list or in the form of attribute-value pairs, such as distinct first name or last name attributes. Therefore, for a data record R with identifying attributes $(a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n$ a record in the database consists of the tuple $\langle P_R, \{I_{a_1}, \ldots, I_{a_n}\}, E_e(R)\rangle$ for the simple list case and $\langle P_R, \{A_1 : I_{a_1}, \ldots, A_n : I_{a_n}\}, E_e(R)\rangle$ for the attribute-value approach.

The simple search operation in the scheme of Bao et al. allows us to easily introduce complex search queries like logical *AND* ($\wedge$), *OR* ($\vee$), or *NOT* ($\neg$) expressions of simple search queries of the form *record d_i contains keyword w* (for keyword lists) or *attribute a of record d_i is equal to keyword w* (for attribute-value pairs). This can be expressed in the form of a simple grammar:

$$
\begin{aligned}
P \rightarrow & \neg P \\
P \rightarrow & (P \wedge P) \\
P \rightarrow & (P \vee P) \\
P \rightarrow & p(w \in d_i.w) \text{ \textit{(for simple keyword lists)}} \\
P \rightarrow & p(d_i.a = w) \text{ \textit{(for attribute-based keywords)}}
\end{aligned}
$$

These complex queries can be provided by the data source during the `ContructQ()` algorithm and executed by the pseudonymization service when executing the `Search()` algorithm.

Depending on the scenario, the extension to multi-keyword search might require reintroducing the storage of encrypted records. When the scenario demands real search capabilities, for example, to find all patients with the last name *Schmidt*, a data source must be able to look for the desired pseudonym holder in the result set by further attributes. These attributes can be stored encrypted together with the indices and pseudonym in the database as provided by the original scheme of Bao et al.

Since the keyword is just used during generating the symmetric encryption key k which is used for the encryption taking place in index generation, the cleartext is just a random bit string. The pseudonymization service just receives a blinded version of $h_s(w)$ with a distinct blinding value r_w for each keyword, so that even equal keywords result in different indices. Therefore, the pseudonymization service does not learn anything about the keywords.

3.3.3 Extending the Scheme with Fuzzy Search Capabilities

Building up on the extension of the scheme to multiple keywords, we can even provide *fuzzy search capabilities* – searching in which results must not match the keyword but just resemble it to some extent. This extension can prove helpful in the case of typos or inconsistent spellings for identifying attributes, for example, when the pseudonym for an individual is requested using differing spellings of the last name like *Schmidt* and *Schmitt* by different data sources.

One method for fuzzy SE is provided by Li et al. [Li+10] and is based on so-called *wildcard-based fuzzy sets*. These serve as a representation of possible deviations of a word according to the edit distance[10], that is, all words obtainable by deleting, inserting, or substituting a character of the word. Such a fuzzy set is achieved by creating multiple variants of a keyword through substituting one character at a time by a wildcard character `*` and inserting the wildcard character at all possible positions in the keyword. For example, for the keyword `abd` the wildcard-based fuzzy set is represented by

$$S_{abc} = \{\texttt{*abc}, \texttt{*bc}, \texttt{a*bc}, \texttt{a*c}, \texttt{ab*c}, \texttt{ab*}, \texttt{abc*}\}.$$

The fuzzy set for the search term `abD` (which contains a typo) is created in the same manner and results in

$$S_{abD} = \{\texttt{*abD}, \texttt{*bD}, \texttt{a*bD}, \texttt{a*D}, \texttt{ab*D}, \texttt{ab*}, \texttt{abD*}\}.$$

Searching for a match through comparison would be successful because of the match `ab*`.

To use this technique in a SE scheme, before index construction the fuzzy keyword sets S_w for the keywords w of a document are generated. Afterwards the same computations as for single keyword index generation are performed for each fuzzy keyword set member individually. The resulting index set is transferred to the server. Searching for the keyword w' is achieved by creating the fuzzy keyword set $S_{w'}$ for w' and sending the trapdoors for each of the members of the set to the server. The search procedure than checks all keyword-index pairs for matches. At least one match indicates similarity between keyword and document keyword.

We transfer this fuzzy search technique to our solution. Based on the multi-keyword variant detailed in Section 3.3.2 the extension to fuzzy search is straightforward. During index construction via `GenIndex()` we construct an index for all members of the fuzzy keyword set S_w for the identifying attribute w. Similarly, the construction of the query via `ContructQ()` is performed for each member of the fuzzy keyword set for the respective attribute value.

Through fuzzy search, result sets with multiple potential pseudonyms can emerge. To allow a data source to differentiate these results and to find the correct pseudonym, we employ the full protocol of Bao et al., including the storage of encrypted records. During the `Setup()` algorithm the symmetric encryption key e is generated and each data source receives this key from the UM via the `Enroll()` algorithm. When performing the `Write()` operation, a data source provides the search indices and the encrypted record containing all details of an individual required for identifying the matching record for a query. After receiving all results for a query via the `Search()` algorithm, the data source decrypts these records and looks for the matching record and pseudonym. If no matching result is found, they ask the pseudonymization service to create a new pseudonym for provided indices and encrypted records.

The downside of this approach consists in harming *attribute confidentiality* regarding other pseudonyms. Another option would be to always just return the best matching result, for example, based on the number of matching fuzzy set records, like some

[10] More specific, this variant of the edit distance is called the *Levenstein* distance [Lev66] with a fixed maximum distance $d = 1$.

related work does it (cf. Rohde et al. [Roh+21]). But this alternative can potentially lead to false positives, in other words, pseudonyms being incorrectly used for multiple individuals.

3.3.4 Limiting the Linkability

Limited linkability is a way to balance the tradeoff between *data minimisation* and *linkability*. The limitation can be a way to mediate between data demands and increasing re-identification risks through linking more data records. Similar to the approach taken in PEEPLL, we can limit the linkability in our scheme in two ways – with respect to time periods and by using budget accounting.

There are two possible approaches for limiting the time period t, in which pseudonyms are valid. For the first approach, when moving from time period t to $t+1$, the pseudonymization service can simply delete its pseudonym database. Alternatively, it stores the current database at another location and uses a fresh database, if pseudonym disclosure is a requirement (see Chapter 4 for further details to and a technical solution for pseudonym disclosure). Afterwards, when a data record R is searched for (by a single keyword, multiple keywords, or fuzzy search), a previously existing pseudonym P_R^t for this record cannot be returned and a new pseudonym P_R^{t+1} must be created. This allows a limitation of validity time periods without affecting any data sources, the user manager UM, or any used keys or parameters. The second approach consists in letting the UM set up the system in a fresh state via the `Setup()` algorithm and enrolling every data source again. Since relevant keys change, data records and respective pseudonyms created in the former time period t cannot be queried in time period $t+1$. By combining both approaches, we can achieve what Zimmer et al. [Zim+20] call *anytrust*. If only one party – UM *or* pseudonymization service – acts according to protocol, we still achieve the limited linkability property.

Limiting the linkability by budget can be achieved by adding a counter c to each database record. Each time when a query result contains a data record R, so that a data source uses the respective pseudonym P, the counter c is incremented. The simplest case is an unweighted count with a constant budget C. For this case, the counter c is incremented by 1 for each usage until $c \geq C$. Afterwards, the corresponding pseudonym P is no longer returned in requests. Depending on the scenario, more complicated budget limitations are imaginable. By incrementing the counter c not by a fixed value but depending on the query or data record and by using record-dependent budgets C_i, we can achieve flexible budget accounting. In the case of query result sets with multiple results (for example, in the case of fuzzy search), we have to increase all counters of the results and the budget only represents an upper use limit. We have not found a simple way to design this limitation in a way that fulfills anytrust. Anytrust would require some sort of global knowledge of usage counts of a pseudonym accessible to every data source, but data sources in our setting are organized independently without communication between them. All ideas of sharing knowledge between them would require this communication and complicate our architecture.

3.4 Adversary Model

Based on this utilization of the SE scheme, we can present our adversary model for the full protocol. The extensions of the protocol in Sections 3.3.2 to 3.3.4 extend the capabilities of the protocol, but do not change the validity of this adversary model. We assume the semi-honest (also referred to as *passive* or *honest-but-curious*) adversary model [PMB14], in which adversaries do not deviate from a given protocol but try to learn as much information as possible from messages legitimately received during the protocol execution. In our opinion, this is a valid assumption since the scenario inherently dictates trust being placed in the data sources to some extent (for example by requiring them to provide valid data and using the correct pseudonym). Even though a malicious pseudonymization service could provide invalid pseudonyms, in this chapter we focus on the confidentiality of identity data (with regards to various properties, see Section 3.2.2) but not the integrity of research data. Furthermore, we assume secure connections between all parties. An adversary in the position to observe message contents would learn the provided pseudonyms and with that would be able to violate properties like *re-use indistinguishability*. Finally, we assume the well-established ciphers which we use as building blocks for our scheme to be practically secure when used with keys of adequate length.

The collusion of parties is a valid scenario in the semi-honest adversary model [EKR18]. Our protocol is suspectible to some combinations of colluding parties and our scheme does not protect against the following threats. If pseudonymization service and user manager collude, the pseudonymization service would be able to learn specific user key material qk_u. This is used for the preparation of search queries in the `ConstructQ()` algorithm. The pseudonymization service in possession of this material would be able to perform brute-force or dictionary attacks (see Section 2.4.3) against potential identity data id by just performing the `ConstructQ()` algorithm with the key material and comparing the result with a received query. This would violate our property *attribute confidentiality*. A collusion between pseudonymization service and an adversarial data source would allow a similar attack. The data source would be able to perform the brute-force attack by running search queries for potential identity data id until the targeted pseudonym is member of the result set in the `search()` algorithm. While an adversarial data source would generally also be able to perform this attack without colluding with the pseudonymization service, they would need to learn the target pseudonym somehow and be able to perform a lot of search queries without being detected (or rate limited). The collusion of adversarial data sources does not present a real advantage to an adversary since all data sources have the same access to the search interface and the key material of multiple scheme users does not increase the adversary capabilities. The only minor advantage for an adversary would be the distribution of brute-force attacks and a slightly more complex detection of these attacks.

3.5 Implementation and Performance Evaluation

We have implemented our scheme in Python using *Charm*[11], a library for fast prototyping of cryptosystems. Each party (data source, pseudonymization service, and UM) provides a representational state transfer (REST) interface for performing necessary operations.

For the pairing-based cryptography we use a super singular curve with a 512-bit base field provided by Charm. For the encryption of records in the `Write()` operation we chose an authenticated encryption with additional data (AEAD) cipher to provide confidentiality and authenticity for the records. For the encryption during the generation of indices in the `GenIndex()` operation (and the respective decryption in the `Search()` operation) we use AES in ECB mode and random elements $r \in \mathcal{M}$ with a size equal to the block size of AES (128 bit). While ECB is a bad choice in general, in our scheme we only utilize it to encrypt or decrypt completely random, insensitive individual blocks, which are just employed to determine search results.

We measured the computational performance for our single-keyword as well as fuzzy search schemes with respect to write and search operations. These are the operations executed regularly during creating and searching for pseudonyms, so they determine the practical applicability of the scheme in our setting. For the write operation we included the index generation `GenIndex()` and for the search operation the query construction `ConstructQ()` to achieve measurements for the complete operations. We omit measurements for the multi-keyword extension (see Section 3.3.2) because it would yield the same performance (in the attribute-based keyword setting) or a constant multiple of the performance (in the simple keyword list setting) in comparison to the single-keyword protocol. All measurements were performed on an off-the-shelf laptop using an Intel Core i7-6600U CPU with 2.6 GHz and 20 GB RAM.

The single-keyword search depicted in Figure 3.5 requires a search time proportional to the database size since for each database record a single index has to be checked. This results in a quite performant scheme, in which a query, even for a large database size $|D'| = 100.000$, needs less than 0.5 s. The write operation is independent of the database size and a single write operation takes 0.005 s.

Additionally, we have inspected the computation time distribution for the individual instructions in the search operation. For a database size $|D'| = 1$ the pairing operation $k' = h(\hat{e}(q_u(w), ComK_u))$ requires nearly 89 % of the computation time and the decryption of indices just about 6 %. But the pairing operation has to be performed only once for a single-keyword search, while the decryption operation is required for each record in the database. In our experiments, the operations even out and take the same amount of computation time for a database size $|D'| = 90$. In larger databases, symmetric decryption takes up the larger share of the computation time. Increasing the performance of the scheme in scenarios with large number of created pseudonyms therefore should focus on increasing the speed of symmetric crypto operations.

For the fuzzy keyword search, we have to consider not only the database size $|D'|$ but also the length of keywords w because the amount of generated indices and required search tokens depends on this length. For the following measurements, we always use fully random keywords and search tokens, so that a search hit is improbable for

[11] The Charm repository is available at https://github.com/JHUISI/charm (visited on 24.03.2023).

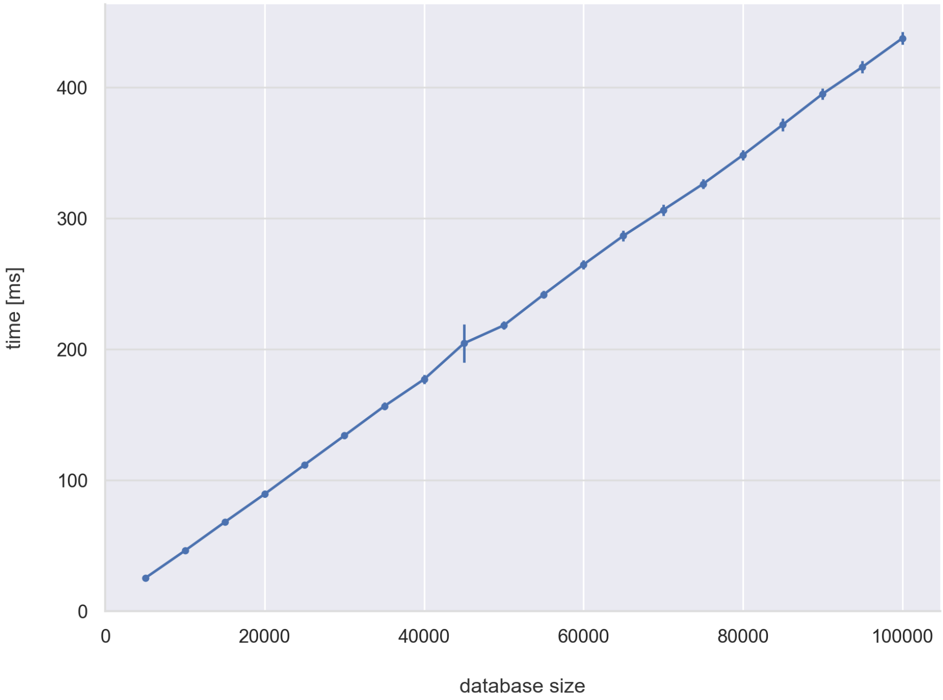


Figure 3.5: Computation times for performing the single-keyword search operation for varying database sizes. The error bars present the standard deviation for 100 runs.

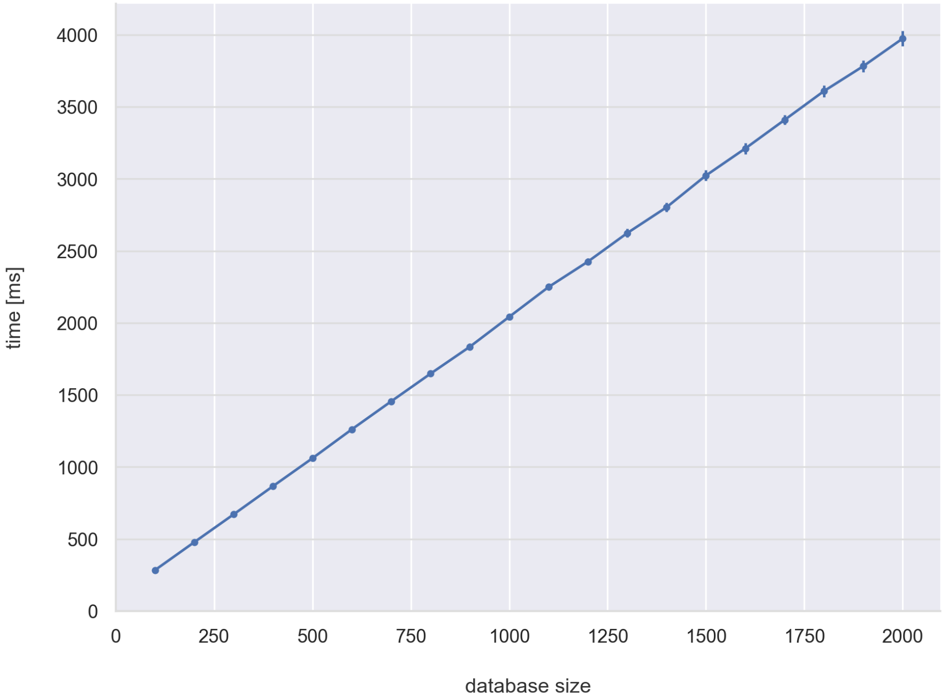


Figure 3.6: Computation times for performing the fuzzy-keyword search operation for varying database sizes and a fixed word length $w = 10$. The error bars present the standard deviation for 100 runs.

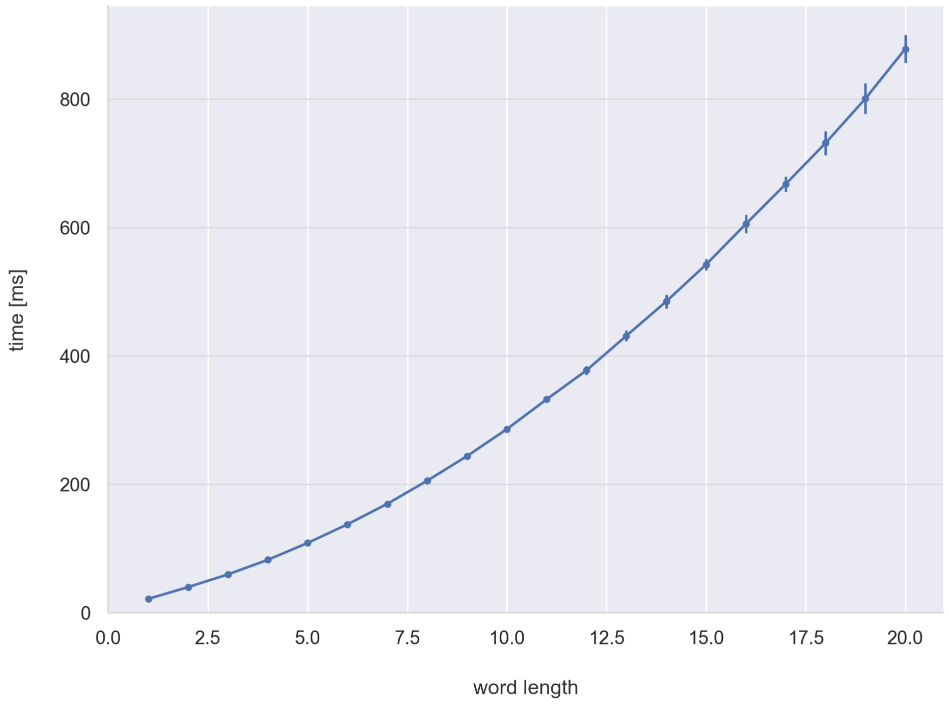


Figure 3.7: Computation times for performing the fuzzy-keyword search operation for a fixed database size $|D'| = 100$ and varying word lengths. The error bars present the standard deviation for 100 runs.

larger word lengths. The complete database including all indices has to be processed. Figure 3.6 shows the required computation time for a fixed word length and varying database sizes. While we see a linear trend, the required time is way higher than in the single-keyword case. This is expected since each database record in our fuzzy scheme contains multiple indices and has to be compared against multiple search tokens. This relationship is observable Figure 3.7, in which the search times for varying word length are presented. Since the number of search tokens as well as indices per database record is proportional to the word length, the computation time is quadratic in the word length.

For the fuzzy keyword search, the computation time for the write operation also depends linearly on the word length as shown in Figure 3.8. A word of size n requires $2n + 1$ indices and for each of these indices the server has to perform the computationally expensive pairing operation $e_w = \hat{e}(r_w \cdot h_s(w), ComK_u)$.

In conclusion, the single-keyword scheme should perform well even for scenarios with hundreds of thousands of records and frequent queries. The fuzzy keyword search scheme might be too computationally expensive even for medium-sized scenarios (depending on the word length at hand). Approaches to increase the performance of the scheme include improving the performance of the symmetric decryption operation and parallelizing the search operation, as all data records can be independently checked for search hits.

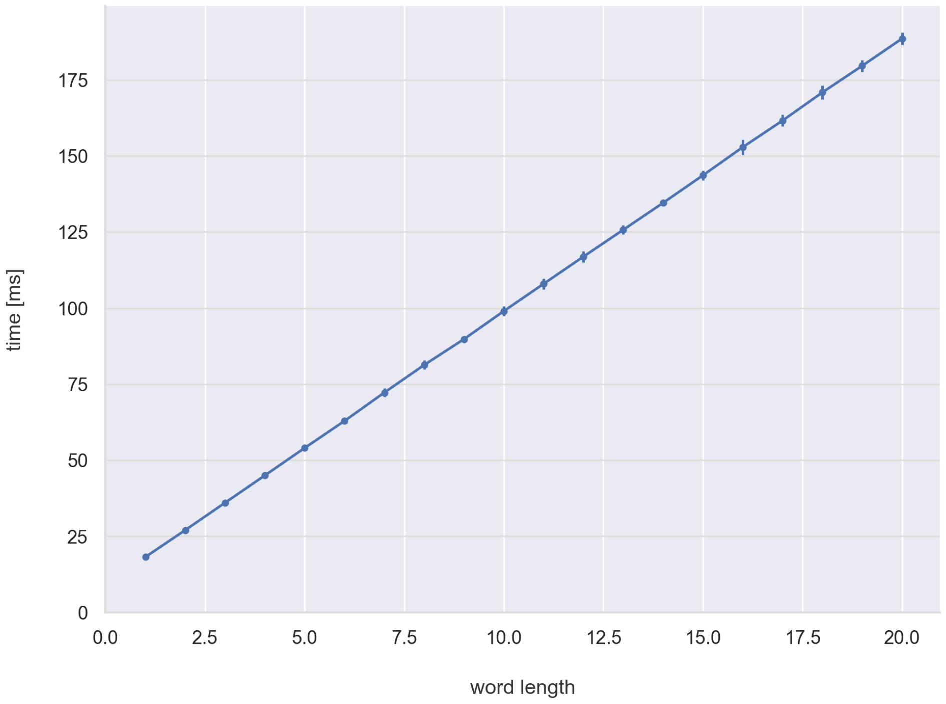


Figure 3.8: Computation times for performing the fuzzy-keyword write operation for varying word lengths. The error bars present the standard deviation for 100 runs.

3.6 Related Work

The problem studied here is related to the concept of privacy-preserving record linkage (PPRL) [Gko+21; Vat+17] – the problem of finding data records from different data sources referring to the same individual in a way that does not leak personal data of respective individuals to the party executing the linking process. But in comparison, PPRL solutions generally focus on one-time matching of datasets and not the incremental setting of this chapter in which data records must be pseudonymized at different points in time [Roh+21]. In the following, we compare our solution to related work with respect to the properties detailed in Section 3.2.2.

3.6.1 PEEPLL

Zimmer et al. [Zim+20] introduce the pseudonymization framework PEEPLL. They utilize a combination of different techniques to achieve different configurations of the system each fulfilling a different subset of all of their proposed protection goals. A HMAC-based construction is employed to prevent the central party from learning the real value of identifying attributes. Each data source is equipped with a key k and sends $HMAC_k(a)$ for attribute value a to the pseudonymization service (which does not possess k). By comparing this value to stored values, an existing pseudonym with the same HMAC value can be returned. Furthermore, PEEPLL can use *Secure Indexes* [Goh03] to hide the requested pseudonym from the pseudonymization service and a simple oblivious transfer (OT) protocol to prevent a data source from being able to perform dictionary attacks against HMAC values potentially belonging to another data source. Finally, PEEPLL allows for limited linkability in form of temporal as well as budget limitation.

PEEPLL achieves *Attribute Confidentiality* by employing the HMAC based comparison of attribute values. There are different instantiations of the framework which fulfill *Re-Use Indistinguishability*, *Matching Pseudonym Unobservability*, and *Limited Linkability* based on different means such as Secure Indexes, OT, and budget accounting. However, the combination of *Matching Pseudonym Unobservability* and *Re-Use Indistinguishability* is not supported. Additionally, PEEPLL does not support *Fuzzy Search Capabilities*. The framework can handle multiple data sources by equipping all of them with a fixed symmetric key, but their enrollment and revocation is not considered.

3.6.2 Mainzelliste

Mainzelliste introduced by Lablans, Borg, and Ückert [LBÜ15] is an open-source[12] software that provides a pseudonymization service for medical data in the form of a REST interface. Clients ask the service for pseudonyms using unaltered identifying data, such as full names and dates of birth. The service searches for an existing pseudonym in the stored data with the help of a configurable record linkage algorithm. Since data is provided in cleartext, some fuzziness (for example, typos) in the data can be tolerated if a error-tolerant record linkage algorithm like, in the case of Mainzelliste, *EpiLink* [Con+05] is applied. If there already is an existing pseudonym for the received identifying data, this pseudonym is returned. Otherwise the service creates a new random pseudonym and returns it.

Mainzelliste performs simple similarity comparisons between received and stored cleartext attribute values during the pseudonymization process. Therefore, it does not fulfill *Attribute Confidentiality* and *Matching Pseudonym Unobservability*. *Re-Use Indistinguishability*, on the other hand, is achieved since the pseudonymization request always provides the data source with the correct pseudonym independent from whether it has existed beforehand. *Limited Linkability* is no design goal and not covered (just as for the Mainzelliste extensions covered below). Since values are transmitted in cleartext, *Fuzzy Search Capabilities* are easily achieved. Multiple data source are directly supported and are managed by simple authentication measures.

3.6.3 Optimized Mainzelliste based on PPRL

Rohde et al. [Roh+21] provide an update to the Mainzelliste software based on ideas from PPRL. They employ *bloom filters* [Blo70] during record linkage instead of the comparison of plaintext identifying data following the approach of Schnell, Bachteler, and Reiher [SBR09]. The basic idea is to partition a data attribute, store all the resulting parts in a bloom filter, and compare similarities of resulting filters during the linkage process. Equal filters indicate the same attributes and consequently data records referring to the same individual. This solution also allows for a fuzzy comparison of data records

[12] The repository can be found at `https://bitbucket.org/medinfo_mainz/mainzelliste/src/master/` (visited on 13.03.2024).

since small deviations in attributes lead to small deviations in the bloom filters due to the partitioning of attributes. The problem of multiple matches with a high similarity is solved by simply choosing the record with the largest similarity score (based on the number of matching bloom filter entries). They present an additional blocking approach to reduce the number of required comparisons and to increase the performance of their solution. However, bloom filter-based PPRL has shown to be vulnerable for some time [Kuz+11; Kuz+12; Nie+14; KS14b; Chr+17]. The attacks are more sophisticated variants of simple dictionary attacks[13] in which the public encoding process is performed for lists of potentially included attribute values and the results are compared to stored bloom filters .

With respect to our properties, the updated version of the *Mainzelliste* employs bloom filters to hide the identifying attribute values and therefore fulfills *Attribute Confidentiality* – at least to some extent due to the vulnerability to dictionary attacks. *Re-Use Indistinguishability* is fulfilled just like in the regular Mainzelliste. This is achieved by always treating the best match above some threshold as the correct one and creating a new pseudonym if none exists. If multiple similar records are present, this approach can lead to a higher false positive rate and incorrectly linked records. Based on these thoughts, it directly follows that *Matching Pseudonym Unobservability* is not fulfilled since the pseudonymization service learns the matching pseudonym. Due to the way the Bloom filters are constructed, the solution provides *Fuzzy Search Capabilities*. Just as for the original Mainzelliste, multiple data source are directly supported and managed by simple authentication measures.

3.6.4 MainSEL

Stammler et al. [Sta+22] introduce the *Mainzelliste SecureEpiLinker (MainSEL)*[14] to prevent the weaknesses of the bloom filter-based approach by Rohde et al. [Roh+21] by using SMPC. Their basic idea is to compute the similarities of bloom filters and respective best matches between data records from two different origins through 2-party SMPC. This approach does prevent adversaries from learning and utilizing the real bloom filter values. They use the SMPC framework *ABY* [DSZ15], a mixed protocol framework for *2-PC*. Because of this, linking records from multiple data origins requires the protocol to be performed between all origins respectively. MainSEL introduces a third party called linkage service which receives and combines the shares resulting from the SMPC protocol and distributes the resulting pseudonyms (or creates new ones in case of no matching records).

MainSEL hides the identifying attributes by using SMPC to compute the similarities of Bloom filters. Therefore it prevents the vulnerability to dictionary attacks present in the solution of Rohde et al. [Roh+21] and fulfills *Attribute Confidentiality*. *Re-Use Indistinguishability* is fulfilled as well because the introduced linkage service hides

[13] This is also related to the comment on the unsuitability of hashing as a pseudonymization technique in Section 2.4.4. Due to the fairly small size of typical identifying attribute domains, such as names, these attacks become practically feasible.

[14] The repositories for *MainSEL* can be found at `https://github.com/medicalinformatics/SecureEpilinker` (SMPC node) and `https://github.com/medicalinformatics/MainzellisteSEL` (central component) (visited on 13.03.2024).

the information whether a pseudonym is re-used or generated. *Matching Pseudonym Unobservability*, on the other hand, is not achieved because the linkage service can pinpoint the matching pseudonym, if any. Just like the version of Rohde et al. [Roh+21] MainSEL supports *Fuzzy Search Capabilities* through their Bloom filter construction. Since MainSEL employs an SMPC framework for two parties, direct support for multiple parties is not provided. Supporting multiple parties could be achieved by letting a party perform the protocol with each other party on a one-by-one basis. But due to the high communication and computation costs of SMPC protocols (cf. Section 5.4), this would impose large performance losses, supposably rendering the solution impractical for multiple parties. Also for this reason, managing the enrollment or revocation of individual data source is not dealt with.

3.6.5 ScrambleDB

Lehmann [Leh19] presents *ScrambleDB*, a pseudonymization concept which does not provide pseudonyms based on identifiers but handles the pseudonymized dataset storage and the data dissemination. The concept introduces four entities. *Data sources* upload their sensitive personal datasets. A *converter* splits the received dataset into multiple datasets, one for each data attribute, which include attribute and identity-specific pseudonyms. A *data lake* stores these unlinkable (“scrambled”) datasets. Finally, *data processors* can receive joined versions of the datasets containing only attributes they are interested in. By employing a novel cryptographic construction, namely a 3-party oblivious and convertible pseudorandom function (PRF), the solution achieves several useful properties. Apart from the data source, no party has access to the original identifier of a data record. Data sources and converter do not learn the pseudonyms stored in the data lake. During the joining of datasets on the request of a data processor the converter and data lake remain unaware of which attributes belong to the same individual. Furthermore, this join is non-transitive, meaning that multiple joins cannot be correlated to deduce more information about individuals. These properties only hold under a non-collusion assumption of converter, data lake, and data processor.

A clever combination of the mentioned PRF and blinding allows ScrambleDB to hide the identifiers from all parties (apart from the data sources) and therefore to fulfill *Attribute Confidentiality*. In comparison to related work presented before and our work, ScrambleDB does not output pseudonyms to the data sources but handles all the data in the data lake and forwards it on request of the data processors. Data sources do not learn any pseudonym, so that *Re-Use Indistinguishability* is fulfilled as well. Furthermore, ScrambleDB also achieves *Matching Pseudonym Unobservability* again due to the employed PRF technique. Additionally, *Limited Linkability* is fulfilled via the distinct join operation which allows for combining parts of the collected data tailored to the needs of the data processor, but not beyond. On the other hand, ScrambleDB does not support *Fuzzy Search Capabilities* and the deterministic PRF-based approach rules out any easy extension for supporting these. Finally, multiple data sources are supported, but ScrambleDB provides no procedure for managing the authorization of these data sources at all.

Table 3.1: Comparison of our work to related work with respect to the properties AC (Attribute Confidentiality), RUI (Re-Use Indistinguishability), MPU (Matching Pseudonym Unobservability), LL (Limited Linkability), FSC (Fuzzy Search Capability), MulDS (Multiple Data Sources), and ManDS (Managable Data Sources) covered in Section 3.2.2.

Approach	AC	RUI	MPU	LL	FSC	MulDS	ManDS
PEEPLL [Zim+20]	✓	(✓)	(✓)	✓	*x*	✓	*x*
Mainzelliste [LBÜ15]	*x*	✓	*x*	*x*	✓	✓	✓
Mainzelliste with Bloom filters [Roh+21]	(*x*)	✓	*x*	*x*	✓	✓	✓
MainSEL [Sta+22]	✓	✓	*x*	*x*	✓	(*x*)	*x*
ScrambleDB [Leh19]	✓	✓	✓	✓	*x*	✓	*x*
This work	✓	*x*	*x*	✓	✓	✓	✓

3.6.6 Comparison to our Work

Our solution accomplishes *Attribute Confidentiality* by employing the SE scheme by Bao et al. [Bao+08] and its property of *query privacy* in particular (see Section 3.1.3). Because data sources have to actively store new entries if a query does not result in a matching pseudonym, *Re-Use Indistinguishability* is not achievable with our solution. *Matching Pseudonym Unobservability* is also not realized in general since the pseudonymization service learns the result set for a search query in which pseudonyms are contained. *Limited Linkability* can be achieved (detailed in Section 3.3.4) as well as *Fuzzy Search Capabilities* (described in Section 3.3.3). Furthermore, our solution supports multiple data source and also allows for actively managing allowed data source through the `Enroll()` and `Revoke()` algorithms and the protocol property of *revocability* (see Section 3.1.3). A summarizing comparison to related work is depicted in Table 3.1.

In summary, these considerations make it clear that there currently is not a perfect solution to the problem of globally consistent pseudonyms that achieves all of our properties. Depending on the constraints specified by the application scenario, such as necessary security properties and number of parties, a suitable solution must be selected.

3.7 Conclusion

In this chapter we have presented a scheme for globally consistent pseudonyms which allows to link data records from various data sources in a privacy-preserving manner without relying on a trusted party which learns all identity-pseudonym relationships. It employs a SE scheme supporting multiple readers and writers introduced by Bao et al. [Bao+08]. In comparison to the work of Zimmer et al. [Zim+20] this scheme enables us to manage authorized data sources, in other words, enrolling or revoking data sources or change keys in case of lost or disclosed keys. We have provided variants of the scheme for deriving pseudonyms from single attributes, multiple attributes, and even similar versions of attribute values (fuzzy keyword search). Furthermore, we presented ideas to limit the linkability of data records with respect to time or budget restrictions. The limitation can be a way to mediate between data demands and increasing re-identification risks through linking more data records. An implementation has been used to evaluate the practical performance of the scheme. Finally, we have compared our scheme to related work in terms of several proposed properties.

One drawback of our scheme is that our approach for fuzzy search is computationally expensive, especially since it does depend on the length of identifiers. Using a bloom filter based approach like the one utilized by Rohde et al. [Roh+21] would be expected to reduce the search times substantially. Whether it also achieves practical accuracy for linking the correct identifiers without many false positives remains to be investigated.

As we have seen, currently there is no scheme – ours included – which fulfills all of our proposed properties. Depending on the constraints dictated by the application scenario, one must choose the most suitable scheme. Future research might look further into practical, privacy-preserving schemes for globally consistent pseudonyms.

Despite continuous developments in privacy-preserving computations, such as in SMPC (see Section 5.1) and *federated learning* (see Section 2.7.6), and their expanding application in privacy-preserving data analysis without central data collection (whether pseudonymized or in cleartext), pseudonymization still has its place. Its central role in the GDPR underscores this importance. Furthermore, pseudonymization is the only technique which directly enables the re-identification of the respective data subject when disclosure is necessary. In Chapter 4 we present an approach to protect this sensitive disclosure process.

4 Pseudonym Disclosure based on the Multi-Eye-Principle

In some situations, it is a necessity to re-identify pseudonym holders, also referred to as the disclosure of pseudonyms. For example, the disclosure of patient identities for pseudonymized medical research data is required when the patient must be contacted in case of incidental findings in studies, for data quality management (cf. Section 6.1.3), or for consultative participation [Pra21]. On the other hand, disclosing the identity behind a pseudonym can have severe consequences for data subjects, especially in domains like medicine – consequences which pseudonymization should prevent in the first place. So the decision about a disclosure should not be made carelessly and disclosure by individual adversaries should be prevented at all cost. For this reason, our goal in this chapter is to enforce the *multi-eye principle* for pseudonym disclosure.

The *multi-eye principle*, a specific variant of the *separation of duties* concept, is a control measure in environments with high security requirements. The principle states that critical actions, decisions, or processes require the approval of two or more authorized persons. The aim of the principle is to reduce the risk of errors or misconduct like theft, fraud, or misuse of information. Other terms used for this principle include *four-eyes principle* (*4EP*), *two-person rule*, and *dual control principle*. It is compulsory in several areas including the access to bank vaults, corruption prevention in public administration[1], the administration of computers systems[2], and as far as the launch of nuclear weapons.

The main idea of this chapter is to employ *threshold cryptography* to distribute the pseudonym disclosure process across multiple parties to enforce the multi-eye principle. Threshold cryptography, a concept established in the 1980s and 1990s, deals with distributing the capability to perform cryptographic operations across multiple parties. This is accomplished by requiring collaboration of a specified minimum number of parties to perform the operation, achieved through the distribution of involved secret key material. Thus, this approach enables the distribution of trust among multiple parties. There are schemes for distributing the signing operation in cryptographic signature schemes [Sho00; Bol02] (*threshold signatures*). Other schemes allow to distribute the decryption operation in public-key encryption schemes [DF90] (*threshold decryption*). We focus on the latter of the two in this chapter and transfer it to the problem of pseudonym disclosure.

Basic schemes in the area of threshold cryptography have been known for years. But the practical application of cryptographic schemes does not only depend on the scheme

[1] For example, the multi-eye principle is a central measure mentioned in the *Federal Government Guidelines for the Prevention of Corruption in the Federal Administration* (in german *Richtlinie der Bundesregierung zur Korruptionsprävention in der Bundesverwaltung*) accessible at `https://www.verwaltungsvorschriften-im-internet.de/bsvwvbund_30072004_04634140151.htm` (visited on 25.03.2024).

[2] For example, in the *IT-Grundschutz-Kompendium* [BSI23] of the Federal Office for Information Security (BSI) the principle is mentioned for the administration of systems with increased need for protection (in german *erhöhter Schutzbedarf*).

itself. Especially questions of key management are highly relevant. This is a non-trivial problem for regular public-key schemes already, but it gets even harder when dealing with multiple parties and shared keys in threshold schemes. Apart from lost or compromised keys, we now have to deal with extra tasks like joining or resigning parties. An example in the business environment is the adjustment of the scheme in case of new or terminated employees.

In the context of threshold decryption it is not trivial to change authorized parties, because a message is encrypted for a specific set of parties and the distributed decryption operation is performed at a later point in time. The set of authorized parties might change between encryption and decryption of a message. This leads to the problem of updating the composition of authorized parties in a secure manner, such that existing ciphertexts can be decrypted by the new set of parties afterwards. There are several approaches for updating shares to deal with certain reasons for changing authorized parties, but they all suffer from disadvantages.

In this chapter we introduce a novel approach for updating key shares in a threshold decryption setting. The main idea is the application of PRE, a technique that translates ciphertexts encrypted with the public key of one key pair to be decryptable with a secret key of another key pair. In our setting, these schemes allow to transfer the ability to decrypt ciphertexts encrypted for a set of parties to another set of parties.

Our main contributions are the following:

- We recap an approach to enforce the multi-eye principle for pseudonym disclosure based on threshold decryption.
- We propose a novel scheme for updating threshold decryption key shares.
- We instantiate our approach with existing schemes for threshold decryption and bidirectional, multi-hop PRE and provide a way to compute the proxy key in a distributed way.
- We develop a library, which implements this scheme, and evaluate the performance and therefore the practical applicability of our scheme.

The chapter is structured as follows: In Section 4.1 we describe our notation and fundamental techniques on which we build our work. Section 4.2 presents our approach to pseudonym disclosure based on the multi-eye principle by utilizing threshold decryption. In this section we just provide an abstract approach without instantiating it with a specific threshold decryption scheme for a cleaner presentation of the idea. We explain different scenarios in which cryptographic keys in the scheme have to be updated and properties an update process might entail in Section 4.3. An overview of existing approaches for these updates is provided in Section 4.4. In Section 4.5 we describe our update solution in detail and also introduce a scheme instantiation with specific cryptographic schemes. Section 4.6 presents the attacker model for the pseudonym disclosure approach based on our update solution. We cover implementation details and a performance evaluation in Section 4.7 and conclude this chapter in Section 4.8.

The idea of using threshold decryption as a method for enforcing the multi-eye principle (recapped in Section 4.2) was developed in my master's thesis [Pet+19]. The use of elliptic-curve cryptography (ECC) and distributed key generation (DKG) protocols in

Section 4.2 and the remaining parts of the chapter dealing with updating cryptographic key material are novel to this work.

4.1 Background

In this section we clarify the notation that we use and provide a brief overview of fundamental techniques that our work relies on.

We use the term *party* for the entities involved in the described protocols. Related work sometimes refers to a party as participant, share-holder or share-owner. When talking about the i'th party we denote this as P_i. We denote an access structure, the family of sets of parties able to perform a threshold decryption operation (see also Section 4.2), as capital letter, such as A. The value x related to access structure A is denoted as $x^{[A]}$.

For computations performed in finite prime fields of order p we omit the required modulo operation $\pmod p$ for cleaner presentation if the situation is unambiguous. An elliptic curve over a finite field $\mathbb{F}_p$ is denoted as E and we use P as a generator point of order q in a large subgroup of the curve. We use additive operation notation and denote curve points by uppercase letters and multiplicative scalars by lowercase letters. In the public-key setting we use $d \in \mathbb{Z}_q$ as the secret scalar and $Q = dP$ as the public key point.

4.1.1 Shamir's Secret Sharing

Shamir's secret sharing [Sha79] is a method to split a secret s into n shares so that a threshold t of shares is required to reconstruct the secret. Less than t shares reveal no information about the secret at all, establishing information-theoretic security. The scheme works as follows. Let p be prime and all calculations performed in $\mathbb{Z}_p$.

- For a secret $s \in \mathbb{Z}_p$ choose a polynomial $f(x) = s + a_1 x + \cdots + a_{t-1}x^{t-1}$ with $a_1, \ldots, a_{t-1}$ chosen randomly from $\mathbb{Z}_p$.
- The evaluations $f(x_i)$ of this polynomial for distinct values $x_1, \ldots, x_n$ form the n shares of the secret.
- The secret s can be reconstructed from any t shares $f(x_i)$ via Lagrange interpolation

$$s = \sum_{i=1}^{t} f(x_i)\lambda_i$$[3]

  using the Lagrange coefficients $\lambda_i = \prod_{j=1, j\neq i}^{t} \frac{-x_j}{x_i - x_j}$.

[3] We slightly overload the notation here, since indices are re-assigned potentially in comparison to their initial generation.

4.1.2 Threshold Decryption

Threshold decryption is a technique which allows to distribute the decryption operation of a public-key scheme over n parties in possession of shares of the secret key in a way that t parties are required to interact for decryption. The parameter t is the basis for referring to these schemes as threshold schemes, as it defines the minimum threshold that must be met to enable the execution of the operation. It allows to mediate between trust distribution and safety requirements in the sense that multiple but not all parties are required to perform the operation. In the decryption operation, parties compute so-called partial decryptions using their respective shares of the private key, which then can be combined to yield the plaintext. During this process the private key sk is not disclosed to any party and less than t partial decryptions do not reveal the plaintext for a given ciphertext encrypted with public key pk. The following definition is based on [BS20].

Definition 4.1.1. A **public-key threshold decryption scheme** consists of four efficient algorithms $(G_{TD}, E, D_{TD}, C_{TD})$:

- $G_{TD}(n, t, \sigma) \rightarrow (pk, sk_1, \ldots, sk_n)$: The key generation algorithm G_{TD} takes a security parameter σ and outputs a public key pk and n secret key shares $sk_1, \ldots, sk_n$ depending on the threshold parameter t.
- $E(pk, m) \rightarrow c$: The encryption algorithm E takes the public key pk and the plaintext message $m \in M$ and outputs the ciphertext c.
- $D_{TD}(sk_i, c) \rightarrow \widetilde{c}$: The partial decryption algorithm D_{TD} takes a secret key share sk_i and the ciphertext c and computes a partial decryption $\widetilde{c}$.
- $C_{TD}(c, \widetilde{c}_1, \ldots, \widetilde{c}_t) \rightarrow m$: The combine algorithm C_{TD} takes the ciphertext c and t partial decryptions $\widetilde{c}_1, \ldots, \widetilde{c}_t$ computed with t distinct private key shares and outputs the plaintext message m corresponding to c.

4.1.3 A Threshold Decryption Scheme

Desmedt and Frankel [DF90] developed one of the first threshold decryption schemes based on the ElGamal public-key scheme [Elg85] and Shamir's secret sharing [Sha79]. Koblitz [Kob87] describes a version of the ElGamal scheme based on ECC, which we adapt here for the given threshold decryption scheme. The scheme works as follows:

Let E be an elliptic curve over a finite field $\mathbb{F}_p$ and P a generator point of order q in a large subgroup of the curve. During key generation each party P_i receives their share $(x_i, y_i) = (x_i, f(x_i))$ of the randomly chosen secret key $d \in \mathbb{Z}_q$ using the approach of Shamir's secret sharing with the polynomial $f(x) = d + a_1 x + \cdots + a_{t-1} x^{t-1}$ for randomly chosen $a_1, \ldots, a_{t-1} \in \mathbb{Z}_q$. Furthermore, the public key $Q = dP$ is constructed. This step can be performed by utilizing a trusted third party or in a distributed manner without a trusted third party (cf. Section 4.2).

The encryption of a message M is performed just like in the ElGamal scheme by computing the ciphertext $(C_1, C_2) = (rP, rQ + M)$ for a random element $r \in \mathbb{Z}_q$. The partial decryption $\widetilde{C}_i$ is computed by party P_i as $\widetilde{C}_i = y_i C_1$. At least t partial decryptions can be combined by an arbitrary party to reveal message m using the Lagrange coefficients λ_i for secret reconstruction: $M = C_2 + (-\sum_{i=1}^{t} \lambda_i \widetilde{C}_i)$. The correctness of this construction is easy to validate:

$$\begin{aligned}
&C_2 + (-\sum_{i=1}^{t} \lambda_i \widetilde{C}_i) \\
=&C_2 + (-\sum_{i=1}^{t} \lambda_i y_i r P) \\
=&rQ + M + (-(\sum_{i=1}^{t} \lambda_i y_i) r P) \\
=&rdP + M + (-drP) \\
=&M
\end{aligned}$$

4.1.4 Distributed Key Generation

While threshold decryption schemes allow parties in possession of shares of a private key to collaboratively decrypt ciphertexts, they do not deal with the problem of how to generate these shares in the first place. This generation can be achieved by a central trusted party which creates the keys and distributes them to the parties. However, in a lot of scenarios where distributed trust is an objective such a party does not exist. This challenge can be tackled by DKG schemes which allow a set of parties to jointly compute the public key and a set of secret key shares without disclosing the private key to any party (as long as there are enough non-colluding parties). The first DKG scheme was published by Pedersen [Ped91] and an improved version preventing an attack manipulating the secret key distribution was introduced by Gennaro et al. [Gen+99]. In the following, we present the scheme of Pedersen to clearly showcase the basic principles in these schemes and not hide them between additional safeguarding protocol steps as it would be the case in the scheme of Gennaro et al.

For parties $P_1, \dots, P_n$, an elliptic curve E over a finite field $\mathbb{F}_p$, and a generator point P of order q in a large subgroup of the curve the scheme works as follows.

- Each party P_i chooses $x_i \in \mathbb{Z}_q$ at random and computes $h_i = d_i \cdot P$. Additionally, they choose a random string r_i and broadcast a commitment $C_i = C(h_i, r_i)$.
- When all parties have exchanged their commitments, they broadcast the values h_i and r_i and check the validity of all commitments. Afterwards, the public key $h = \sum_{i=1}^{n} h_i$ can be computed.

At this point, the public key h as well the (virtual) private key $d = \sum_{i=1}^{n} d_i$ are already set. But the private key can only be computed based on each and every share d_i – the shares represent a (n, n) sharing of d). The remaining steps are performed to achieve a (t, n) sharing of the private key so that only t shares are required to recompute the private key.

- Each party P_i chooses a polynomial $f_i(z) = d_i + f_{i_1}z + \cdots + f_{i_{t-1}}z^{t-1}$ with random coefficients $f_{i_k} \in \mathbb{Z}_q$. This polynomial has degree $t-1$ and $f_i(0) = d_i$.
- For the coefficients of this polynomial each party P_i computes $F_{ij} = f_{ij} \cdot P$ $(j \in \{0, \ldots, t-1\})$ and broadcasts these values.
- Each party P_i sends the values $s_{ij} = f_i(j)$ secretly to P_j. They keep the value s_{ii} to themselves.
- Each party P_i verifies the received s_{ji} values by checking $s_{ji} \cdot P \overset{?}{=} \sum_{l=0}^{t-1} F_{jl} \cdot i^l$.
- If all checks are successful, each party P_i computes their private share $s_i = \sum_{j=1}^{n} s_{ji}$ and signs the public key h to indicate its validity (using some signature scheme independent of the key generation scheme).

In a final verification step all parties can review the validity of the key generation.

- For this purpose each party P_i broadcasts $\sigma_i = s_i \cdot P$.
- Any party can then check that $\sigma_i \overset{?}{=} \sum_{j=1}^{n}(h_j + \sum_{l=1}^{t-1} F_{jl} \cdot i^l)$ based on the broadcasted values h_j and F_{jl}.

In the end, each party P_i is in possession of a share of the private key s_i and the parties have jointly computed the respective public key h.

4.1.5 Proxy Re-Encryption

Let (pk_i, sk_i) and (pk_j, sk_j) be two key pairs for an asymmetric encryption scheme. PRE is a technique for converting ciphertexts encrypted with public key pk_i to ciphertexts being decryptable with sk_j. In other words: PRE allows to re-encrypt ciphertexts for a different key pair. The re-encryption can be performed by a third party, the proxy, using a so-called proxy key computed from both key pairs or a subset of these keys. During this operation the proxy does not learn any plaintext or any of the involved private keys. The principle of PRE was introduced by Blaze, Bleumer, and Strauss [BBS98] and they also proposed the first PRE scheme based on a modified version of the ElGamal cryptosystem.

There are two properties of PRE schemes relevant for this paper:

- **bidirectional/unidirectional:** This property describes if the re-encryption with one proxy key can just be performed in one or in both directions, that is, only from sk_i to sk_j) or vice versa as well [Ate+06]. The same property is sometimes also referred to as symmetry/asymmetry [BBS98].
- **single-/multi-hop:** Single-hop schemes just allow a single re-encryption of a ciphertext, whereas multi-hop schemes allow further re-encryptions of already re-encrypted ciphertexts [CH07].

Further properties, like proxy invisibility or collusion safeness, which are not directly relevant to this work, are described in more detail by Ateniese et al. [Ate+06]. A comprehensive overview of further PRE research is provided by Qin et al. [Qin+16]. In this paper only multi-hop schemes are of interest, therefore we give the following definition based on [MS17].

Definition 4.1.2. A multi-hop PRE scheme consists of five probabilistic polynomial time algorithms $(G_{PRE}, E, D_{PRE}, RG_{PRE}, RE_{PRE})$:

- $G_{PRE}(\sigma) \rightarrow (pk, sk)$: Given a security parameter σ, the key generation algorithm G_{PRE} outputs a key pair pk, sk.
- $E(pk, m) \rightarrow c$: The encryption algorithm E computes a ciphertext c from message m and public key pk.
- $D_{PRE}(sk, c) \rightarrow m$: The decryption algorithm D_{PRE} takes a secret key sk and ciphertext c and returns the message m.
- $RG_{PRE}(sk_i, pk_i, sk_j, pk_j) \rightarrow \pi_{i\rightarrow j}$: The proxy key generation algorithm RG_{PRE} takes two key pairs pk_i, sk_i and pk_j, sk_j and computes the proxy key $\pi_{i\rightarrow j}$.
- $RE_{PRE}(c_i, \pi_{i\rightarrow j}) \rightarrow c_j$: The re-encryption algorithm RE_{PRE} takes the proxy key $\pi_{i\rightarrow j}$ and a ciphertext c_i encrypted with pk_i and translates it to ciphertext c_j decryptable with sk_j. If the scheme is bidirectional, RE_{PRE} can be used with $\pi_{j\rightarrow i}$ to translate a ciphertext c_j to c_i as well.

4.1.6 A Bidirectional Multi-Hop Proxy Re-Encryption Scheme

A bidirectional multi-hop PRE scheme based on the ElGamal public-key encryption scheme is introduced by Ivan and Dodis [ID03]. The already mentioned ECC-based version of the ElGamal scheme by Koblitz [Kob87] is adapted for this scheme as well. The scheme works as follows:

Let E be an elliptic curve over a finite field $\mathbb{F}_p$ and P a generator point of order q in a large subgroup of the curve. Party P_i chooses a secret key $d_i \in \mathbb{Z}_q$ and computes their public key $Q_i = d_i P$. The encryption of a message M works just as in the regular ElGamal scheme. Select a random element $r \in \mathbb{Z}_q$ and compute the ciphertext as $(C_1, C_2) = (rP, rQ_i + M)$. To decrypt a ciphertext, party P_i computes $M = C_2 + (-d_i C_1)$.

Given another key pair d_j, Q_j for Party P_j, the proxy key can be generated as $\pi_{d_i \leftrightarrow d_j} = d_j - d_i$. To perform the re-encryption, the C_2 component of a ciphertext encrypted with Q_i has to be updated so that the public key Q_j is included instead and Party P_j can decrypt the ciphertext. Component C_1 remains unchanged. To re-encrypt ciphertext component C_2 one computes

$$\begin{aligned}
C_2' &= C_2 + \pi_{d_i \leftrightarrow d_j} C_1 \\
&= rQ_i + M + (d_j - d_i) rP \\
&= rQ_i + M + r d_j P + (-r d_i P) \\
&= rQ_i + M + rQ_j + (-rQ_i) \\
&= rQ_j + M
\end{aligned}$$

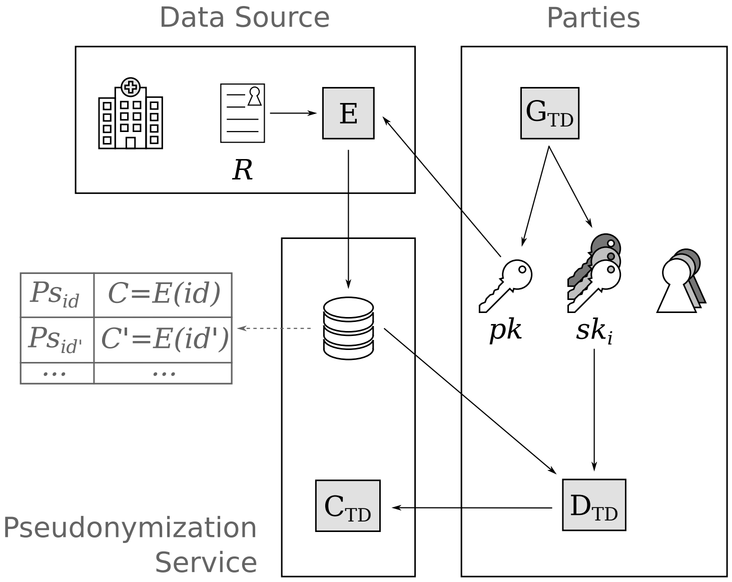


Figure 4.1: The high-level overview of pseudonym disclosure via threshold decryption.

4.2 Pseudonym Disclosure Protected by Multi-Eye Principle

The main idea of this chapter is to employ *threshold decryption* as a technical measure to achieve the multi-eye principle for pseudonym disclosure. Details about threshold decryption schemes are presented in Section 4.1.2. In the following we describe the pseudonym generation and disclosure processes utilizing threshold decryption on a scheme-agnostic level. An instantiation with a specific threshold decryption scheme will be given in Section 4.5. We stick to the system model and parties (data source, data processor, and pseudonymization service) introduced in Section 3.2.1 – even though the general method is applicable to further pseudonymization scenarios. The high-level idea of this chapter is depicted in Figure 4.1.

First, the (t, n) threshold decryption scheme has to be initialized by performing the key generation algorithm G_{TD}. During this process a public key pk as well as key shares sk_i for the n parties responsible for decryption are generated. This setup can be achieved by using a central trusted party. However, in a lot of scenarios where distributed trust is an objective such a party does not exist. Therefore, we focus on the case where the parties perform this setup interactively via a DKG scheme. These schemes allow a set of parties to jointly compute the public key and a set of secret key shares without disclosing the private key itself to any party (as long as there are enough non-colluding parties). Details about DKG schemes are provided in Section 4.1.4. The schemes require secure (authenticated and encrypted) point-to-point channels between the parties, so the existence of these channels is assumed from now on. Approaches on how to establish these channels are provided by Aumasson, Hamelink, and Shlomovits [AHS20].

When a data source in possession of the public key pk pseudonymizes a data record R containing identifying information id, apart from substituting id with a pseudonym Ps_{id} obtained from the pseudonymization service (see Chapter 3), the identifying information id is also encrypted with the encrypt algorithm E of the threshold decryption scheme using the public key pk. The resulting ciphertext C is stored together with Ps_{id} in the mapping table of the pseudonymization service[4]. Afterwards, the pseudonymized data record R' is sent to the data processor.

When the data subject behind the pseudonym Ps_{id} has to be disclosed on the data processor's instructions, a qualified subset of all parties perform the partial decryption algorithm D_{TD} using their respective key share sk_i and the ciphertext C forwarded by the pseudonymization service. The family of sets of parties, which are able to perform this cryptographic operation is referred to as the *access structure* (sometimes also *committee* [Mar+19]) of the scheme. For (t, n) schemes this family is given by all sets of parties with at least t members. Note that simple (t, n) schemes can easily be generalized to more complex access structures [ISN89]. The partial decryptions $\widetilde{C}_i$ resulting from the partial decryption can then be combined by employing the partial decryption combination algorithm C_{TD} (performed by a dedicated party or the pseudonymization service) to restore the identifying information id. Finally, these information are sent to the data processor. Since less than t parties are not able to restore these information, this approach cryptographically enforces the multi-eye principle for pseudonym disclosure.

In comparison to simple approaches, however, key management tasks in distributed protocols like threshold decryption schemes entail additional complexity. There are situations which require updating the party composition or shares of the parties in an access structure. More formally, for access structures A and B with respective public keys $pk^{[A]}$ and $pk^{[B]}$ and underlying $(t^{[A]}, n^{[A]})$ and $(t^{[B]}, n^{[B]})$ threshold decryption schemes, we want to perform an update operation, so that ciphertexts originally encrypted with $pk^{[A]}$ can afterwards be decrypted by combining $t^{[B]}$ partial decryptions computed using shares from access structure B[5] In the remainder of this chapter we focus on how to perform access structure updates. We provide relevant situations for and properties of update approaches, review existing approaches, and present our own solution. Since the remainder of this chapter is not specific to pseudonym disclosure anymore, we stick to terms and parties used in threshold decryption or proxy re-encryption literature (party, proxy, client) in the next sections for clarity.

4.3 Situations and Properties

There are several situations in which we have to change the access structure of the scheme. Common situations we focus on in this chapter are adding a party (also referred to as *enrollment*), removing a party (also referred to as *disenrollment*), increasing or decreasing the threshold parameter t, lost key shares, compromised key shares, proactive key rotation, and transferring ciphertexts to a completely different access structure. Some of these situations can be handled by custom-tailored protocols. However, we look for a generic solution, which is applicable to all given situations.

[4] When used for other pseudonymization techniques (see Section 2.4.4) the ciphertext might have to be stored in a distinct database. If reversible techniques for pseudonym creation (for example, symmetric encryption) are used, it must be ensured that this reversibility does not undermine the multi-eye principle.

[5] When we informally mention the public key of an access structure or speak about decryption being performed by an access structure, it should be understood as described here.

In addition, we propose several properties the access structure update operation should entail.

- Plaintext secrecy (**PS**): Plaintexts of existing ciphertexts remain encrypted.
- Key secrecy (**KS**): Neither the shared private key nor any key share of the old or new access structure is disclosed to non-owners.
- Prevention of previous access structure access (**PAA**): The previous access structure must not be able to perform threshold decryption of ciphertexts after the access structure update.
- Erasure free model (**EFM**): **PAA** holds without requiring parties to delete their keys or key shares. This prevents the old access structure from performing the decryption operation even if enough malicious parties keep their old shares. Therefore, in addition to **PS** and **KS**, this property is central to the practical security of the scheme.
- Update operation without ciphertext access (**UCA**): The update operation can be performed without requiring parties to access ciphertexts. This property enables the separation of ciphertext storage and update operation. Otherwise, if a malicious set of parties is part of the scheme's access structure and these parties need to access the ciphertexts during the update operation, they can decrypt all ciphertexts. An update operation that does not require ciphertext access allows the scheme to recover from this situation.
- Unchanged public key (**UPK**): Performing the update operation does not require to replace the public key. This property is desirable because replacing a public key is associated with (sometimes complex) organizational overhead to distribute the new public key to clients and to ensure its authenticity.

4.4 Existing Approaches for Access Structure Updates

This section covers existing approaches for updating access structures in a threshold decryption scenario and how they fulfill our properties (cf. Section 4.3).

4.4.1 Naïve solution

The simplest solution is to decrypt the ciphertext $c^{[A]}$ or to recover the private key $sk^{[A]}$ by collaboration of at least t parties. Afterwards, the directly or indirectly resulting plaintexts can be encrypted with the new public key $pk^{[B]}$ of a new access structure. This solution would be suitable in all of our situations, but has the major disadvantage that it discloses all plaintexts (and in case of key recovery also the private key) to the combining party. This solution might be suitable when a trusted third party exists (which would have to delete plaintexts after the update operation). However, as stated before, we focus on scenarios where such a party is not present.

4.4.2 Monotonous access structure updates

Ito, Saito, and Nishizeki [ISN89] presents a theoretical approach for monotonous access structure updates, meaning that the new access structure is an extension of the old one. This approach is suitable for adding new parties to the access structure, but not for other situations such as removing parties. Additionally, the approach does not consider practical aspects, such as the distribution of new shares.

4.4.3 Dynamic secret sharing

A class of related approaches can be summarized under the term dynamic secret sharing schemes, which can be employed for secret sharing-based threshold decryption schemes. They extend secret sharing schemes with the ability to update aspects of the scheme, such as changing the threshold or disenrolling parties. These schemes can be divided into three classes based on the underlying communication models and their objective.

Redistributing shares

First, there are approaches [DJ97; MSW99] which achieve dynamic access structures by redistributing shares to the parties of the new access structure via secure channels. These approaches are suitable for all of our situations. They keep the shared secret and update the shares via redistribution between all parties without necessary ciphertext access, but require the parties to delete their old shares. Therefore, they are not able to recover from existing sets of malicious parties in the old access structure.

Updating shares via broadcast

Next, there is a variety of related work [Mar+99; Zha+12b; BJM05; Blu+94; Bla+93] which covers schemes allowing to update the access structure in some predefined ways by just using public broadcast messages after relying on secure channels during initialization. These schemes require a priori knowledge of possible changes (like the maximum number of disenrollments or further thresholds) and are not able to enroll new parties. Furthermore, most approaches keep the original secret and require the deletion of old shares or prohibit their usage.

Proactive secret sharing

Another family of approaches, called *proactive* secret sharing schemes, is concerned with the secure renewal of shares of the original access structure, protecting against mobile adversaries, who compromise parties one-by-one over a period of time. Schemes are introduced by, amongst others, Herzberg et al. [Her+95], Schultz, Liskov, and Liskov [SLL10], and Maram et al. [Mar+19]. These schemes are just usable for some of our situations and furthermore require the deletion of old key shares.

4.4.4 Quorum-controlled approach

A unidirectional PRE scheme based on threshold cryptography is proposed by Jakobsson [Jak99]. The re-encryption of ciphertexts is performed by a quorum of proxy servers, which are in possession of shares of the global secret key. This scheme can be used for access structure updates in our setting. The parties (proxy servers) of the current access structure A can re-encrypt respective ciphertexts for the new access structure B by using just the new public key $pk^{[B]}$. This solution is suitable for all considered situations without disclosing keys or plaintexts. However, for the re-encryption all parties must collaborate for every single ciphertext. Afterwards they have to delete their shares and the processed ciphertexts. In case of a malicious set of parties in the access structure, these parties have access to and can decrypt all ciphertexts during the update operation.

4.4.5 Blinding-based approach

Zhou et al. [Zho+05] design a protocol similar to the previously described quorum-based solution by Jakobsson. They incorporate a distributed blinding protocol and so-called verifiable dual encryption, so that some computations during the re-encryption can be performed by the new access structure or even in preparation. This approach suffers from the same drawback as the one presented before: A collaboration of all parties is required for the re-encryption of all individual ciphertexts, shares and ciphertexts have to be delete after the update operation, and malicious parties are able to decrypt all ciphertexts.

4.4.6 Summary

All existing solutions to the problem of updating access structures suffer from not satisfying relevant properties. Table 4.1 summarizes the different approaches with respect to proposed properties. We discuss stated properties of this work in Section 4.5.6.

4.5 Employing Proxy Re-Encryption for Access Structure Updates

In this section, we introduce a novel scheme for access structure updates. The general idea is to use proxy re-encryption for updating the ciphertexts from access structure A to B. We employ the ideas and algorithms of PRE and threshold decryption, but extend them with the possibility to compute the proxy key in a distributed manner.

PRE schemes require a *proxy key* π generated from the old and new key pair or a subset of these keys. To achieve a secure protocol in the given setting, this key must be created by the parties in a distributed manner interactively without revealing shares or private keys. Our idea is to perform this operation similar to the approach taken in threshold decryption to distribute the decryption operation. A party P_i uses their private key shares for the old and new key pair to compute an intermediary result, we call *partial proxy key* $\widetilde{\pi}^i$. A set of partial proxy keys can be combined to provide the proxy key. Afterwards this key can be used by the *proxy*, which performs the re-encryption without being able

Table 4.1: Overview over existing approaches and their properties. **PS** = non-disclosure of plaintexts for existing ciphertexts, **KS** = non-disclosure of the shared private key or key shares, **PAA** = the old access structure cannot perform the threshold decryption after translation, **EFM** = **PAA** holds without relying on parties deleting keys or key shares, **UCA** = the update operation does not require parties to access existing ciphertexts, **UPK** = the update operation does not require replacing the public key. For approaches which are not suitable for all situations, the covered situations are given: adding a party (**A**), removing a party (**R**), changing the threshold parameter t (**T**), lost key shares (**L**), compromised key shares (**C**), proactive key rotation (**P**), and transferring ciphertexts to a completely different access structure (**D**). Properties in parentheses indicate incomplete fulfillment of properties.

Approach	**PS**	**KS**	**PAA**	**EFM**	**UCA**	**UPK**	**All Situations**	
Naïve threshold decryption	*x*	✓	✓	*x*	*x*	*x*	✓	
Naïve private key recovery	*x*	*x*	✓	*x*	✓	*x*	✓	
Monotonous update	✓	✓	*x*	*x*	✓	✓	*x*	(only **A**)
Redistributing shares	✓	✓	✓	*x*	✓	✓	✓	
Update via broadcast	✓	✓	✓	*x*	✓	✓	*x*	(only **R**, **T**, **L**, **C** & **P**)
Proactive secret sharing	✓	✓	✓	*x*	✓	✓	*x*	(only **C** & **P**)
Quorum-controlled	✓	✓	✓	(*x*)	*x*	*x*	✓	
Distributed blinding	✓	✓	✓	(*x*)	*x*	*x*	✓	
This work	✓	✓	✓	(✓)	✓	*x*	✓	

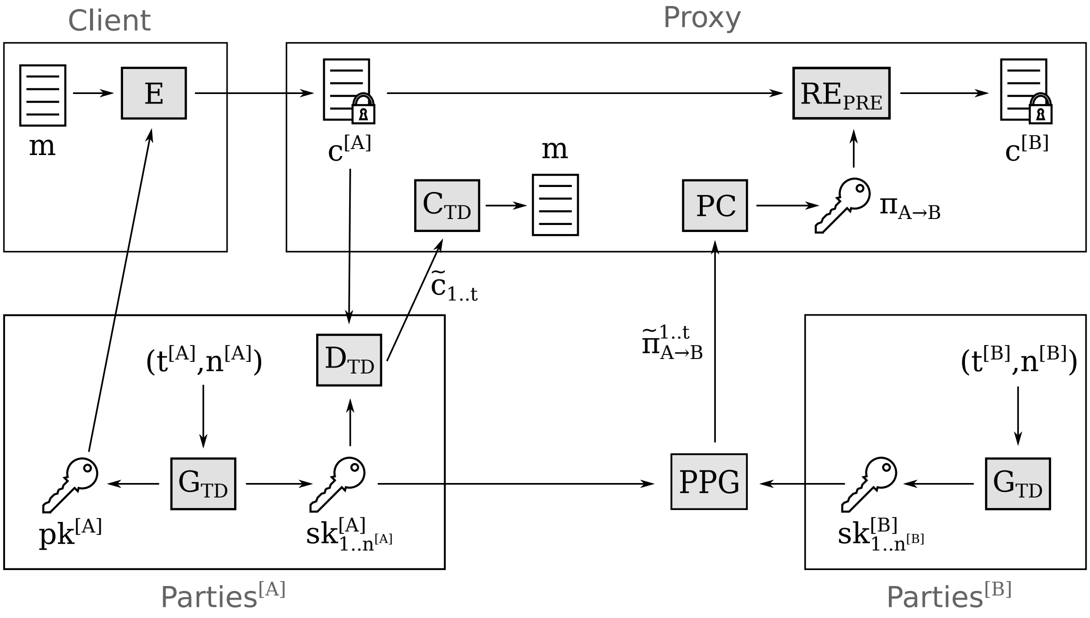


Figure 4.2: Overview of our scheme and interplay of the algorithms, as defined in Definition 4.1.1, 4.1.2, and 4.5.1, including key generation G_{TD}, the generation of partial proxy keys PPG, the combination of partial proxy keys PC, and the re-encryption RE_{PRE} for access structures A and B. For a cleaner presentation the algorithms for encryption E, partial decryption D_{TD}, and combination C_{TD} are only shown for access structure A.

to reveal the ciphertexts, the private key or key shares. An overview of the complete scheme is depicted in Figure 4.2. The following definition formalizes such a scheme by extending Definitions 4.1.1 and 4.1.2 and introducing new operations for the distributed generation of the re-encryption key.

Definition 4.5.1. Our scheme extends the algorithms of Definitions 4.1.1 and 4.1.2 and consists of seven algorithms $(G_{TD}, E, D_{TD}, C_{TD}, PPG, PC, RE_{PRE})$:

- $G_{TD}, E, D_{TD}, C_{TD}$, and RE_{PRE} are the same algorithms as in Definitions 4.1.1 and 4.1.2.

- $PPG(pk^{[A]}, sk_i^{[A]}, pk^{[B]}, sk_i^{[B]}) \rightarrow \widetilde{\pi}^i_{A\rightarrow B}$: PPG takes the private key shares for access structures A and B of party P_i as well as the respective public keys as input and outputs a partial proxy key $\widetilde{\pi}^i_{A\rightarrow B}$.

- $PC(\widetilde{\pi}^1_{A\rightarrow B}, \dots, \widetilde{\pi}^t_{A\rightarrow B}, pk^{[A]}, pk^{[B]}) \rightarrow \pi_{A\rightarrow B}$: PC takes t partial proxy keys as well as the respective public keys as input and outputs the proxy key $\pi_{A\rightarrow B}$.

In the following we present an instantiation of such a scheme, cover its correctness and security, analyze the applicability to the situations given in Section 4.3 and discuss the properties of the scheme with respect to the ones given in Section 4.3.

4.5.1 Scheme Instantiation

In this section, we develop an instantiation of our scheme based on the threshold decryption scheme given in Section 4.1.3 and the bidirectional multi-hop PRE scheme given in Section 4.1.6. Using a multi-hop scheme is a vital requirement for our solution as this allows updating the access structure an unlimited number of times. Using a bidirectional scheme allows reverting the re-encryption process of ciphertexts using the same proxy key. Since we want to prevent the old access structure from performing the decryption operation, it is disadvantageous to be able to reconstruct the ciphertexts for the old access structure after the re-encryption operation (see Section 4.5.6 for further discussion). So ideally we want to use a *unidirectional multi-hop* scheme. However, we have found no way to compute a proxy key in a distributed manner in such a scheme, so we leave this as an open problem for future research.

The instantiation uses variants of the operations G_{TD}, D_{TD}, and C_{TD} from the threshold decryption scheme, RE_{PRE} from the PRE scheme and the encryption operation E which is the same in both schemes. Furthermore, we provide constructions for the novel operations in our scheme – the generation and combination of partial proxy keys, PPG and PC. In the following we present the scheme operations in detail.

Let E be an elliptic curve over a finite field $\mathbb{F}_p$ and P a generator point of order q in a large subgroup of the curve. The parties compute their private key shares (x_i, y_i) for the (virtual) secret scalar $d \in \mathbb{Z}_q^*$ and the public key point $Q = dP$ employing an ECC-based variant of the DKG protocol from [Gen+99].

In the original ECC-based ElGamal scheme [Kob87] the public key Q is used for encrypting a message encoded as curve point M: $(C_1, C_2) = (rP, M + rQ)$ for a random $r \in \mathbb{Z}_q^*$. However, in practice this textbook version of the scheme is insecure [BJN00]. Our scheme therefore uses a hybrid approach based on Elliptic Curve Integrated Encryption Scheme (ECIES) [ABR01]. Further we utilize an authenticated encryption scheme for symmetric encryption, which removes the requirement of an additional MAC in ECIES as proven by Kurosawa and Matsuo [KM04]. Based on this, encryption works as follows: A random value $k \in \mathbb{Z}_q^*$ is chosen and used to compute the point $K = kP$. A key derivation function KDF is employed to derive a symmetric key $k_s = KDF(K)$. This key is used to encrypt the message m with the authenticated encryption scheme AE to compute c_s. The ciphertext has three components $(C_1,\ C_2,\ c_s) = \big(rP,\ K + rQ,\ AE_{k_s}^{enc}(m)\big)$.

A *partial decryption* of the ciphertext is obtained by a party in possession of share (x_i, y_i) by computing $\widetilde{C}_i = y_i C_1$. Combining t partial decryptions discloses dC_1:

$$\sum_{i=1}^{t} \lambda_i \widetilde{C}_i = \left(\sum_{i=1}^{t} \lambda_i y_i \right) C_1 = dC_1$$

This is used to compute the point K:

$$C_2 + (-dC_1) = K + rdP + (-drP) = K$$

This point K is used to derive $k_s = KDF(K)$ and to decrypt the message $m = AE_{k_s}^{dec}(c_s)$.

We now cover the new operations PPG for generating partial proxy keys and PC for combining partial proxy keys to compute the proxy key itself. In particular, we provide a way to perform these operations in a distributed manner without leaking secret keys or key shares.

Let A and B be two access structures for respective $(t^{[A]}, n^{[A]})$ and $(t^{[B]}, n^{[B]})$ schemes. The parties of these schemes have setup their respective (virtual) private keys $d^{[A]}$ and $d^{[B]}$ in the form of key shares $(x_1^{[A]}, y_1^{[A]}) \ldots, (x_{n^{[A]}}^{[A]}, y_{n^{[A]}}^{[A]})$ and $(x_1^{[B]}, y_1^{[B]}), \ldots, (x_{n^{[B]}}^{[B]}, y_{n^{[B]}}^{[B]})$ employing the already mentioned DKG protocol.

Assume that for all different situations (cf. Section 4.3) there exists a shared set of shares in both access structures, the old one A and the new one B. The validity of this assumption will be justified in Section 4.5.5. Each party P_i in this shared set computes their partial proxy key $\widetilde{\pi}_{A\to B}^i = \lambda_i^{[B]} \cdot y_i^{[B]} - \lambda_i^{[A]} \cdot y_i^{[A]}$. The Lagrange coefficients λ_i state which shares are included in the secret reconstruction and are not secret, so they can just be precomputed by the proxy and forwarded to the participating parties. After receiving $t = \max(t^{[A]}, t^{[B]})$ partial proxy keys, the proxy computes the proxy key as the sum of these partial proxy keys $\pi_{A\to B} = \sum_{i=1}^{t} \widetilde{\pi}_{A\to B}^i$.

The re-encryption can afterwards be performed similar to the PRE scheme given in Section 4.1.6. With the proxy key $\pi_{A\to B} = d^{[B]} - d^{[A]}$ the proxy re-encrypts the second component of a ciphertext $(C_1, C_2^{[A]}, c_s)$ (which is the only component dependent on the access structure): $C_2^{[B]} = C_2^{[A]} + \pi_{A\to B} C_1$. The components C_1 and c_s remain unmodified. This is also advantageous from a security and a performance perspective, since in case of an access structure update only a small key component needs to be re-computed, while the authenticated encryption c_s of the (potentially large) message m remains untouched.

An overview of the complete scheme is presented in Table 4.2. A proof for its correctness is given in Section 4.5.2 and arguments for its security in the semi-honest adversary model in Section 4.5.3.

4.5.2 Proof of Correctness

The correctness proofs for the underlying threshold decryption scheme [DF90] (cf. Section 4.1.3) and the PRE scheme [BBS98] (cf. Section 4.1.6) remain valid in our scheme. It remains to show that ciphertexts originally encrypted for access structure A and re-encrypted for a new access structure B can in fact be decrypted by B. The proxy key computation

$$\pi_{A\to B} = \sum_{i=1}^{t} \widetilde{\pi}_{A\to B}^i$$

can be transformed to

$$\pi_{A\to B} = \sum_{i=1}^{t} \lambda_i^{[B]} \cdot y_i^{[B]} - \sum_{i=1}^{t} \lambda_i^{[A]} \cdot y_i^{[A]}.$$

According to Shamir's secret sharing (cf. Section 4.1.1), the secret D with shares $(x_1, y_1), \ldots, (x_n, y_n)$ is reconstructed as $D = \sum_{i=1}^{t} \lambda_i \cdot y_i$ with $\lambda_i = \prod_{j=1, j\neq i}^{t} \frac{x_j}{x_j - x_i}$. In our

Table 4.2: An overview of the full scheme including all of its algorithms.

Algorithm	**Responsible**	**Computations**
Key Generation G_{TD}	Parties	DKG following the steps from [Gen+99] providing the public key Q and n key shares (x_i, y_i).
Encryption E	Client	For message m to be encrypted: $k \xleftarrow{r} \mathbb{Z}_q^*$ $K = kP$ $k_s = KDF(K)$ $r \xleftarrow{r} \mathbb{Z}_q^*$ $(C_1, C_2, c_s) = (rP, K + rQ, AE^{enc}_{k_s}(m))$
Partial decryption D_{TD}	Parties	For P_i holding share (x_i, y_i) and ciphertext (C_1, C_2, c_s): $\widetilde{C}_i = y_i C_1$
Partial decryption combination C_{TD}	Proxy	For ciphertext (C_1, C_2, c_s): $dC_1 = \sum_{i=1}^{t} \lambda_i \widetilde{C}_i$ $K = C_2 + (-dC_1)$ $k_s = KDF(K)$ $m = AE^{dec}_{k_s}(c_s)$
Partial proxy key generation PPG	Parties	For access structure update from A to B and P_i holding shares $(x_i^{[A]}, y_i^{[A]})$ and $(x_i^{[B]}, y_i^{[B]})$: $\widetilde{\pi}^i_{A \to B} = \lambda_i^{[B]} y_i^{[B]} - \lambda_i^{[A]} y_i^{[A]}$
Proxy key combination PC	Proxy	For access structure update from A to B: $\pi_{A \to B} = \sum_{i=1}^{\max(t^{[A]}, t^{[B]})} \widetilde{\pi}^i_{A \to B}$ $\pi_{A \to B} P \overset{?}{=} Q^{[B]} + (-Q^{[A]})$
Re-Encryption RE_{PRE}	Proxy	For access structure update from A to B and ciphertext $(C_1, C_2^{[A]}, c_s)$: $C_2^{[B]} = C_2^{[A]} + \pi_{A \to B} C_1$

threshold scheme this recovers the secret scalars. Since the partial proxy keys $\widetilde{\pi}^i_{A\to B}$ are assumed to be computed from parties in a shared set of both access structures, we obtain $\pi_{A\to B} = d^{[B]} - d^{[A]}$, which is exactly the proxy key of the PRE scheme in Section 4.1.6. During re-encryption the only access-structure-dependent component $C_2^{[A]}$ is transformed, so that the secret scalar $d^{[A]}$ is replaced with $d^{[B]}$:

$$\begin{aligned}
&C_2^{[A]} + \pi_{A\to B} C_1 \\
=&rd^{[A]}P + K + (d^{[B]} - d^{[A]})rP \\
=&rd^{[B]}P + K \\
=&C_2^{[B]}
\end{aligned}$$

Ciphertexts encrypted for access structure A and re-encrypted with $\pi_{A\to B}$ can afterwards be decrypted by utilizing the given threshold decryption process from Definition 4.1.1.

4.5.3 Security Arguments

We use the semi-honest adversary model for reasoning about the security of our scheme. Involved parties do not actively deviate from the protocol derived from the scheme, but are curious to learn all possible information from legitimately received messages.

We argue that knowing a ciphertext (C_1, C_2, c_s), the proxy key $\pi_{A\to B} = d^{[B]} - d^{[A]}$, any number of partial re-encryption keys $\pi^i_{A\to B}$, and less than t partial decryptions do not leak any private key shares, the private key itself or the plaintext.

Due to the semantic security of the employed encryption schemes and relying on the security guarantees from [ID03] and [DF90] we can argue that: Proxy key and ciphertext do not reveal the plaintext, the proxy key does not reveal secret keys $d^{[A]}$ or $d^{[B]}$, less than t partial decryptions do not reveal the plaintext, and partial decryptions do not reveal private key shares.

It remains to show that partial proxy keys $\widetilde{\pi}^i_{A\to B} = \lambda_i^{[B]} y_i^{[B]} - \lambda_i^{[A]} y_i^{[A]} \pmod q$ do not reveal private key shares of the old or new access structure. As already mentioned in Section 4.5.1 the Lagrange coefficients are public values. The key shares $y_i^{[B]}$ and $y_i^{[A]}$ are drawn from a uniform distribution due to the used DKG protocol for all parties. As the partial proxy keys are computed over the finite field $\mathbb{F}_q$, the uniform distribution still holds for $\lambda_i y_i$. An attacker gains no additional information about $y_i^{[A]}$ or $y_i^{[B]}$ from the difference of the two uniformly distributed random values.

We want to emphasize that these arguments do not represent a formal proof of the security of our method, but simply indicate that the intended security requirements are potentially met. A cryptographic security proof could not be provided within the scope of this work and is considered future work.

4.5.4 Extension to Malicious Adversaries

In order to provide guarantees against stronger, malicious attackers, we can enhance our protocol with additional sanity checks. We can rely on guarantees of the used DKG protocol, which uses a verifiable secret sharing scheme [Fel87], to detect inconsistent shares and cheating parties during the key generation phase. Before re-encryption the validity of a computed proxy key can be verified using only the old and new public keys by checking $\pi_{A \to B} P \overset{?}{=} Q^{[B]} + (-Q^{[A]})$. However, detecting which parties behave dishonest during the generation of the proxy key, that is which parties provide an incorrect partial proxy key, would require more sophisticated techniques and is left for future research. Another open problem in the malicious setting is the question of how to prove the honesty of the proxy for the re-encryption step. Techniques like Jakobson's translation certificates [Jak99] or the verifiable proof by Zhou et al. [Zho+05] might be a starting point for research in this direction. But the employed authenticated symmetric encryption via AEAD ciphers at least provides a way to check the authenticity of a ciphertext when decrypting a message during the partial decryption combination operation.

4.5.5 Applicability to Situations

As stated, we require a shared set of parties in the old and new access structure. Furthermore, it is required that a honest set of parties from the old access structure is initially able to perform the decryption operation (which is an important requirement in the case of lost key shares).

Section 4.3 mentions several situations which require an update of the access structure or the key shares. In the following we just consider the situations one-by-one and not in combination. If the scheme is not applicable for combined variations of the situations, the scheme can be applied by iterative executions. We now justify that under all conditions a shared set in the access structures A and B exists (except for access structures differing completely, where a slightly more complicated construction is given):

- **Adding a party:** Any set of parties of A can be used since $A \subset B$.
- **Removing a party:** When removing party P_r, each set of A not containing P_r is still member of B and can be chosen to generate the proxy key. An exception would be an (n, n)-scheme, but this is no reasonable case since afterwards the decryption would not be possible anymore.
- **Increasing the threshold parameter from $t^{[A]}$ to $t^{[B]}$ with $t^{[A]} < t^{[B]}$:** Each set of A with at least $t^{[B]}$ parties can be used. If $t^{[B]} > n$ there exists no such set, but this would render the scheme useless anyway.
- **Decreasing the threshold parameter:** Any set of parties of A can be used since $A \subset B$.
- **Proactive key rotation:** This is the easiest case, since $A = B$. Each set of the access structure can be used.
- **Lost key shares:** $A = B$ holds as well, with the caveat that the owner of the lost share P_l cannot be included in the proxy key generation. If $t = n$ holds for the threshold scheme, there exists no valid set without P_l. In this case, the threshold decryption is not possible anymore.

- **Compromised key shares:** For this situation $A = B$ holds as well with the caveat that the owner of the compromised share P_c should not be included in the proxy key generation. If this is not preventable, it must be assured that the attacker gets no access to the partial proxy key of P_c. Otherwise with the knowledge of the old share and the partial proxy key, the new share can be trivially computed as well.
- **Transferring the ciphertexts to a different access structure with $A \cap B = \emptyset$:** For this situation our scheme is not directly applicable. However, a solution is to perform a two-step approach. First, re-encrypt the ciphertexts to a scheme with access structure $A \cup B$ under collaboration of a set of parties in A. Afterwards re-encrypt the resulting ciphertexts to a scheme with access structure B under collaboration of a set of parties in B.

4.5.6 Properties of the Scheme

What remains is a look at the properties of our new scheme with respect to the ones given in Section 4.3. An overview of the fulfilled properties is given in Table 4.1.

As justified in Section 4.5.3 the provided approach does not disclose keys, key shares or plaintexts during execution of the scheme to unauthorized parties and therefore fulfills our properties **PS** and **KS**.

Since the ciphertexts are re-encrypted for the new access structure (using a new shared key), the old one is not able to perform the decryption of the resulting ciphertexts anymore, which fulfills our property **PAA**.

However, this just holds under the assumption that old ciphertexts are deleted after re-encryption or that less than the threshold t parties of the old access structure keep their shares. In this sense our scheme does not completely fulfill our property **EFM**, but makes it harder to exploit the weakness in comparison to existing approaches, especially the ones based on share redistribution (cf. Section 4.4), by requiring malicious parties to keep their old shares *and* a malicious proxy to keep old ciphertexts. In other words, we distribute the required trust: It is sufficient that the proxy *or* the parties of the old access structure act honestly.

There is one more limitation. Since we instantiate our scheme based on a bidirectional PRE scheme, a malicious proxy in possession of the proxy key and the re-encrypted ciphertexts would be able to reverse the re-encryption process and enable the old access structure to perform threshold decryption again. Using a unidirectional scheme and deleting the old ciphertexts would prevent this malicious action. However, as we would still rely on some deletion taking place at the proxy, this presents just a minor limitation.

Another benefit of our solution compared to related work is that we can recover from a malicious set of parties being able to decrypt ciphertexts at one point in time since the virtual private key changes. Messages encrypted with the new public key cannot be decrypted by the malicious set providing *forward secrecy* (assuming that the malicious set is not a member of the new access structure).

This comes at the cost of a new public key after the access structure update. Therefore our solution does not fulfill our property **UPK**. It might require organizational processes, such as employing existing public key infrastructure (PKI) approaches, to deal with these changes. Another solution would be to keep all proxy keys and update new ciphertexts encrypted with the initial public key on-the-fly (with the problem a bidirectional PRE scheme entails, given above).

Finally, the parties just have to collaborate in order to compute the proxy key once, but are not required to perform operations on all ciphertexts. The re-encryption itself can be performed by any third party acting as the proxy without disclosing plaintexts or keys to this party. This fulfills our property **UCA**. Furthermore, this property is vital for the aforementioned recovery from malicious sets of parties in the access structure. Since parties do not have to access ciphertexts during the update operation, we can prevent malicious access structure sets from decrypting all existing ciphertexts after receiving them in schemes where this access is required.

4.6 Adversary Model for Pseudonym Disclosure

In this section we extend the security considerations with respect to the presented update scheme to an adversary model for the full pseudonymization disclosure process. We assume the semi-honest (also referred to as *passive* or *honest-but-curious*) adversary model [PMB14], in which adversaries do not deviate from a given protocol but try to learn as much information as possible from messages legitimately received during the protocol execution. Even though some steps are taken to extend the scheme to a malicious adversary model (see Section 4.5.4), some key algorithms such as the re-encryption of ciphertexts are not secure in the presence of malicious adversaries. The semi-honest model is nonetheless a helpful assumption in our scenario because it lets us reason about confidentiality aspects in the presence of colluding adversarial parties. Note that the collusion of adversaries is allowed in the semi-honest model [EKR18]. Furthermore, we assume secure point-to-point connections between all parties. Otherwise adversaries would be able to reconstruct key shares and the secret key during execution of the DKG protocol. Finally, we assume the well-established ciphers which we use as building blocks for our scheme to be practically secure when used with keys of adequate length.

The employed threshold scheme provides security against $t-1$ colluding passive adversaries. In our architecture we have separated the decryption process from the storage of ciphertexts. Therefore, even when adversarial parties collaborate with an adversarial pseudonymization service which provides them with access to ciphertexts, less than t adversarial parties are not able to decrypt these ciphertexts (and with that perform pseudonym disclosure).

An adversarial pseudonymization service has the capability to initiate the disclosure process. However, this illegitimate process, in particular the comutation of partial decryptions, requires the participation of at least t adversarial (or at least inattentive) parties. A adversarial minority or even honest parties only prevent the disclosure. Organizational processes which can be used to assess the legitimacy of a disclosure request are considered out-of-scope in this work.

The presented method for updating the access structure of the scheme allows regular updates of the access structure, for example adding or removing parties. But it also allows to recover from a set of adversarial parties in the access structure as long as there are at least t honest parties and an honest pseudonymization service which can successfully perform the update operation. This is possible only because of the separation of proxy key generation from ciphertext storage and re-encryption. If adversarial parties would be able to access ciphertexts they could disclose any pseudonym at will.

As described in Section 4.5.6 our scheme does not completely fufill our property **EFM**. We still require that deletion occurs after the update operation, either by parties deleting their old key shares or by the pseudonymization service deleting old ciphertexts and the proxy key (which is required since we employ a bidirectional PRE scheme).

4.7 Implementation and Evaluation

We have implemented our new scheme in Python[6] in combination with a DKG protocol based on elliptic curves. Where possible existing open-source implementations for cryptographic building blocks were employed. Specifically, we use `pynacl` for authenticated encryption based on the XSalsa20 stream cipher and Poly1305 as MAC for authentication as well as `PyCryptodome` for elliptic curve operations on the National Institute of Standards and Technology (NIST) curve P-256.

To evaluate the performance of our implementation, we performed several measurements on an off-the-shelf laptop using an Intel Core i7-6600U CPU with 2.6 GHz and 20 GB RAM. All parties were simulated on this machine with communication happening locally. We analyze influence of operational parameters, like the used scheme or the message size.

Figure 4.3 shows the measured computation times for the DKG protocol which depend on the utilized (t, n)-scheme. The measurements show that the computation times are influenced by the threshold t as well as the number of parties n. Due to its complexity (see Section 4.1.4), this protocol generally requires the highest computation time by far in comparison to other operations of the scheme, up to several seconds for reasonable access structures. In comparison, Figure 4.4 shows the measured computation times for the centralized key generation operation. Because in this case the keys and shares are computed directly without requiring a sophisticated protocol, the computation times are just in the millisecond range.

In Figure 4.5 the resulting times for encrypting messages E with varying sizes are displayed. It is expected that larger message sizes result in longer computation times since the message is encrypted using an authenticated symmetric encryption scheme whose encryption time grows linearly with the message size. The asymmetric part of the encryption has a constant share on the overall computation time of about 2 ms.

In Figure 4.6 the computation times for combining partial decryptions C_{TD} of encrypted messages with varying sizes are displayed. It is expected that larger message sizes lead to higher operation times since the message is encrypted with an authenticated symmetric encryption scheme whose decryption time grows linearly with message size. The combination of partial decryptions has a constant share on the overall computation time, but depends on the used scheme and in particular the number of required partial

[6] Our open-source implementation is publicly available at `https://github.com/tompetersen/threshold-crypto`.

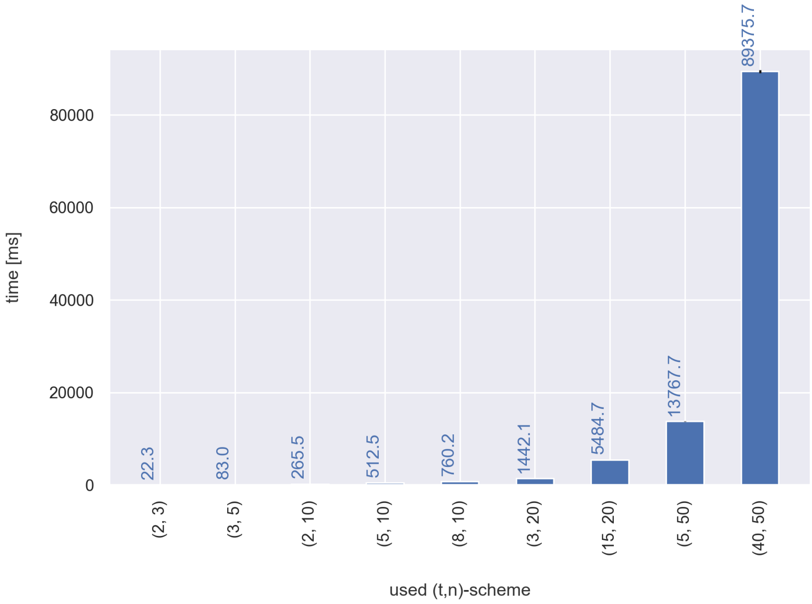


Figure 4.3: Scheme-dependent computation times for DKG. The measured time depicts the arithmetic mean of overall computation times (that is the sum of all parties' computation times). Communication times between parties are not considered. The error bars present the standard deviation for 10 runs.

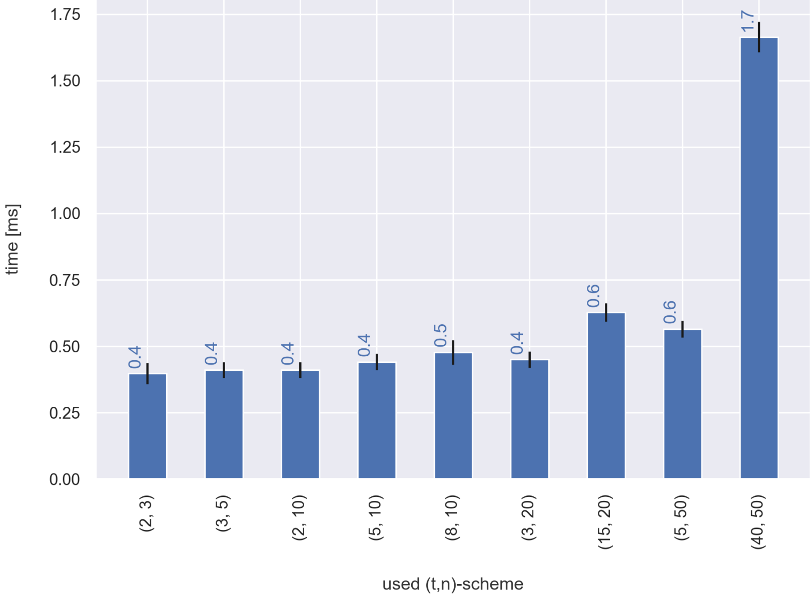


Figure 4.4: Scheme-dependent arithmetic mean of computation times for centralized key generation. The error bars present the standard deviation for 10.000 runs.

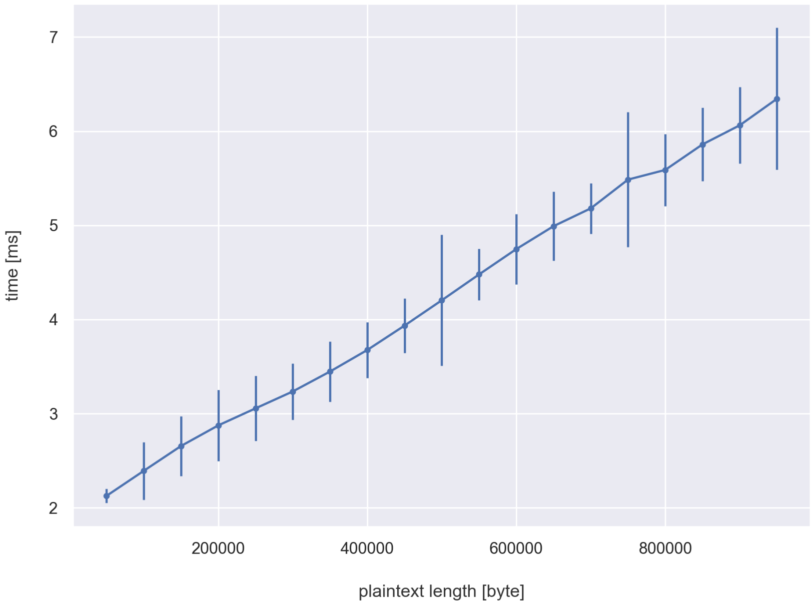


Figure 4.5: Arithmetic mean of computation times for encrypting messages of varying sizes. The error bars present the standard deviation for 10.000 runs.

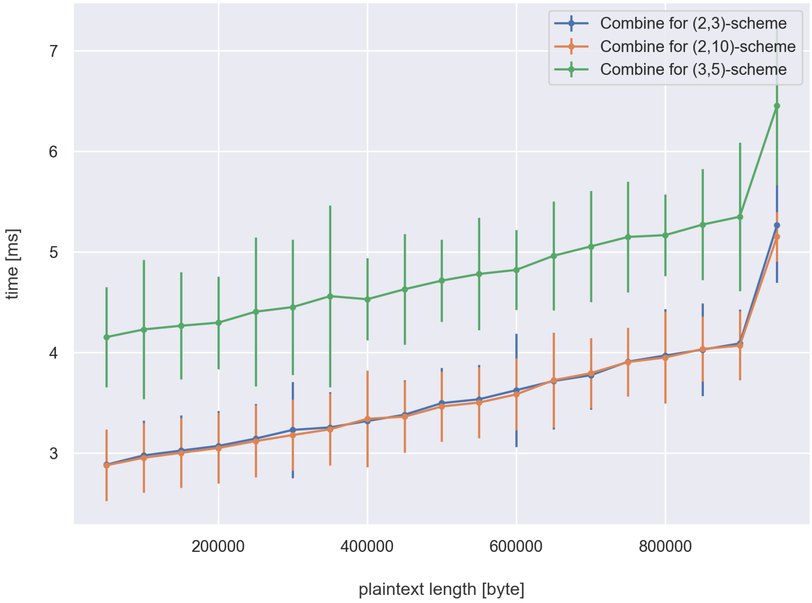


Figure 4.6: Arithmetic mean of computation times for combining partial decryptions for the encryption of messages of varying sizes. In this setup three fixed schemes were used for comparative purposes. The error bars present the standard deviation for 10.000 runs.

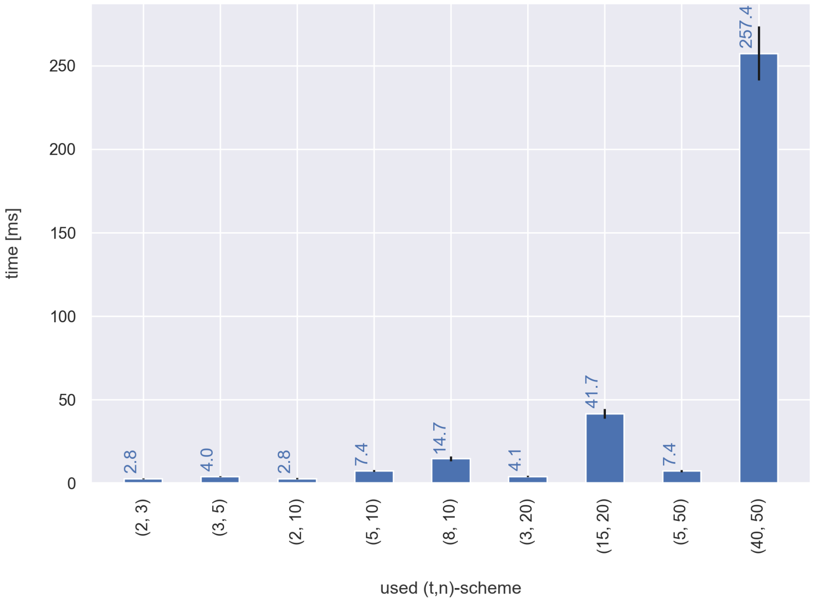


Figure 4.7: Scheme-dependent arithmetic mean of computation times for combining partial decryptions for a fixed encrypted message size of 1.024 bytes. The error bars present the standard deviation for 10.000 runs.

decryptions, that is the threshold t. This correlation is also displayed in Figure 4.7 for various (t, n)-schemes.

Figure 4.8 displays the computation times for the combination of proxy keys PC from partial proxy keys for various (t, n)-schemes. Even though the scheme, in particular the threshold t, influences the computation times, this influence is small. This is because the computation time is dominated by the additional check of the proxy key validity (see Section 4.5.4).

In addition, we measured the computation times required for the remaining operations – the computation of partial decryptions D_{TD}, of partial proxy keys PPG, and the re-encryption of ciphertexts RE_{PRE}. The computation times for these operations are independent of the used scheme as well as the message size.

In practice, some operations such as the decryption of messages would take more time due to network communication, but since most operations require at most one round-trip, this would just add constant time which we omitted in the measurements. The only exception is the DKG, which needs 4 communication rounds between the parties.

In conclusion, the computation times for all of the operations which are expected to be performed on a regular basis and potentially multiple times (encryption, partial decryption, partial decryption combination, and re-encryption) are in the single digit millisecond range for reasonable access structure sizes. The same holds for the computation of partial proxy keys and their combination as well as centralized key generation. The only operation which can take several seconds (depending on the used (t, n)-scheme) is the distributed key generation. But this operation has to be performed only once for each access structure update which is not expected to happen often and should therefore be no obstacle in any practical setting.

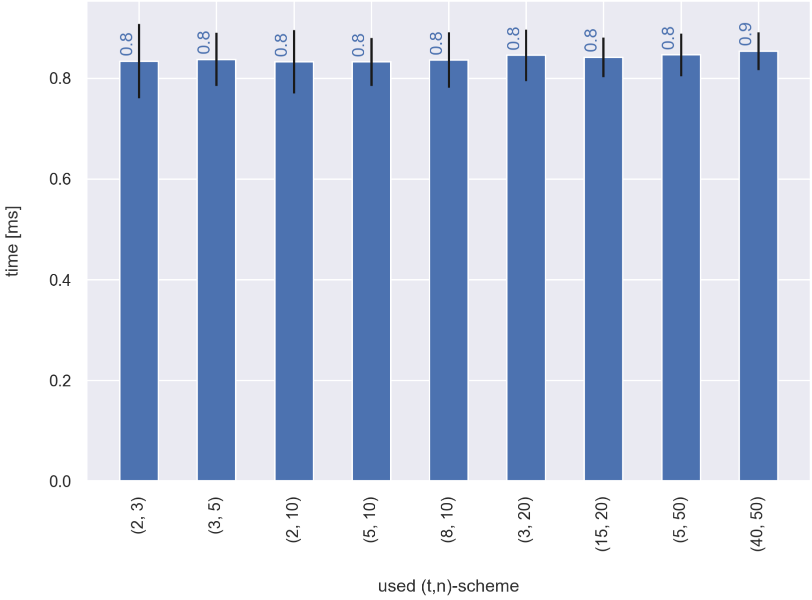


Figure 4.8: Scheme-dependent arithmetic mean of computation times for proxy key combination. The error bars present the standard deviation for 10.000 runs.

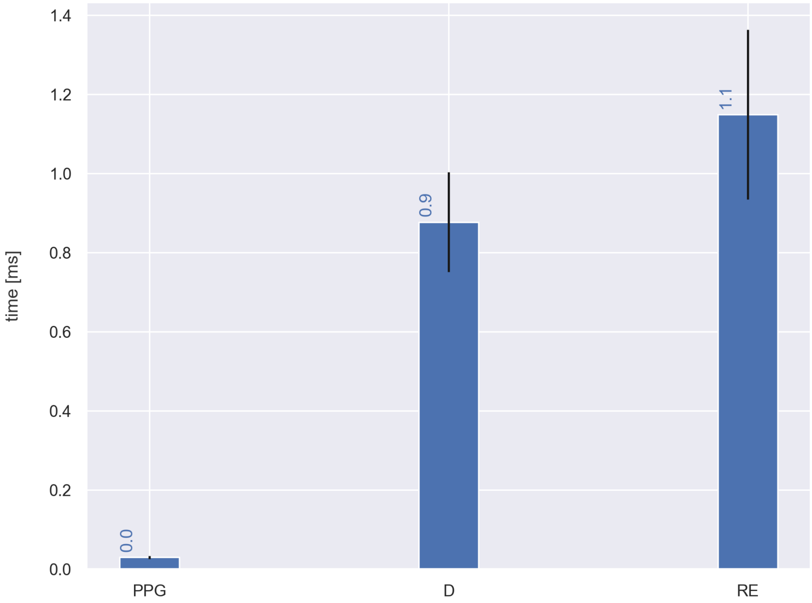


Figure 4.9: Arithmetic mean of computation times for operations (partial proxy key generation, partial decryption, re-encryption) which do not depend on the used scheme or the message size. The error bars present the standard deviation for 10.000 runs.

The sizes of messages sent over the network are similar to the ones known from regular public-key schemes. Namely, the scheme sends less than 1 kB for ciphertexts (apart from the symmetrically encrypted part, which grows linear in the size of the message m), partial decryptions, and partial proxy keys when using common elliptic curves offering 128 or 256 bit of security.

4.8 Conclusion

We have introduced an approach for enforcing the multi-eye principle for pseudonym disclosure based on threshold decryption. While the application of threshold decryption for this purpose is straightforward, key management issues are more complex in these schemes in comparison to simple public-key schemes. We have introduced a novel scheme for updating access structures in a threshold decryption setting based on proxy re-encryption. In comparison to former approaches our scheme ensures secrecy of plaintexts and keys without completely relying on parties deleting their old shares and without requiring an interactive translation of old ciphertexts. We have provided an instantiation of our scheme and implemented it showing its practical applicability. The approach outlined in this chapter may prove beneficial beyond the issue of pseudonym disclosure, serving to protect sensitive data by the multi-eye principle.

Still, there are some limitations and room for future research. In comparison to erasure-dependent approaches we improve the situation in that our scheme does not solely depend on parties deleting their old shares. Instead, it requires enough parties of the old access structure *or* the proxy to act honestly. In this sense, we improve the situation in comparison to erasure-dependent approaches, but still depend on some erasure taking place.

Another limitation of our solution is the fact that the public key changes after the access structure update. It might require organizational processes, such as employing existing PKI solutions, to deal with these changes. Another solution would be to keep all proxy keys and update new ciphertexts encrypted with the initial public key on-the-fly (with the problem a bidirectional PRE scheme entails, given in Section 4.5.6).

There are three major directions for future research:

- We have instantiated our scheme using a bidirectional, multi-hop scheme. Using a unidirectional, multi-hop scheme like [Pho+16] and especially computing the proxy key in a distributed manner in the respective scheme remains an open problem.

- In Section 4.5.3 we covered basic arguments for the security of our scheme. These should not be confused with a full formal security proof, which we do not provide within the scope if this thesis.

- As stated in Section 4.5.3, we use the semi-honest adversary model for our scheme. Giving security guarantees in a malicious adversary model with respect to malicious parties or a malicious proxy server would further increase the usefulness of our scheme. Some indications for potential extensions of the current scheme are provided in Section 4.5.4.

After discussing an architecture for pseudonymization for distributed data sources in Chapter 3 and examining a way to decentralize the pseudonym disclosure process in this chapter, we shift our focus to anonymization methods. When linking data records from different data sources and disclosing data subjects in datasets is not required, it can be a better approach to compute research datasets protected by privacy models in a distributed manner. In Chapter 5 we investigate a specific protocol for the decentralized computation of syntactically private datasets.

5 Distributed Syntactic Privacy

As we have seen in Sections 2.6.9 and 2.7.7 there is no silver bullet in data anonymization. In this sense syntactic privacy models (see Section 2.6) like k-anonymity can still have their merits, even in the presence of stronger privacy models like DP (see Section 2.7). While the syntactic anonymization of locally available data can already be challenging, oftentimes data is held by different parties not necessarily trusting each other. Naive approaches to syntactic anonymization in this context commonly involve two strategies: first, local anonymization of data according to a specified syntactic privacy model followed by the integration of the results (*generalize-then-integrate*), second, the aggregation of all data by a trusted third party that performs the syntactic anonymization locally afterwards (*integrate-then-generalize*) [JX09]. The *generalize-then-integrate* approach is associated with significant utility loss, while the *integrate-then-generalize* method relies on the presence of a trustworthy party, which may not be available in all scenarios. A better approach is to have the parties perform a distributed syntactic anonymization protocol. These protocols result in a dataset which fulfills some syntactic privacy model without sharing more information with other parties than necessary. This approach preserves data utility without requiring a trusted third party.

However, it is a challenging task to design these protocols in a way that parties do not leak additional information. In this chapter we present weaknesses in one of the most-cited distributed syntactic anonymization protocols which, to the best of our knowledge, have not been identified before. Recent advances in the field of general-purpose SMPC frameworks allow us to provide an updated protocol version which fixes these weaknesses while still being practically applicable.

The main contributions of this chapter are the following:

- We identify weaknesses in the distributed syntactic privacy protocol by Mohammed et al. [Moh+10].
- We provide an updated protocol mitigating these weaknesses. It utilizes an SMPC-based subprotocol, which might prove beneficial in situations unrelated to the one of distributed syntactic anonymization.
- We implement the updated protocol employing a recent SMPC framework called *MOTION* [Bra+22] and evaluate the computation and communication demands of our updated protocol on two datasets and with a varying number of parties and QIDs.

The chapter is structured as follows: In Section 5.1 we provide an overview of related work relevant for this chapter. We explain the details of the protocol by Mohammed et al. [Moh+10], their privacy claims, and the shortcomings of the protocol with respect to these privacy claims in Section 5.2. In Section 5.3 we provide an enhanced protocol which mitigates the weaknesses and we evaluate its performance in Section 5.4. Section 5.5 concludes the chapter.

5.1 Background

In this section we provide an overview of distributed syntactic anonymization protocols as well as the field of SMPC – a technique our updated protocol employs to fix the weaknesses of the original protocol.

5.1.1 Distributed Syntactic Privacy

In the field of distributed anonymization one can differentiate between *vertically* and *horizontally* partitioned data [Koh+14]. Vertically partitioned data describes data of individuals being shared between different parties, where each party is in possession of only a subset of all data attributes, while horizontally partitioned data means that each party holds the same set of data attributes for different individuals. An example for anonymizing vertically partitioned data in a distributed manner is provided by Jiang and Clifton [JC06]. We focus on horizontally partitioned data anonymization in this paper.

Jurczyk and Xiong [JX09] introduce a protocol for the anonymization of horizontally partitioned data. It is based on Mondrian (see Section 2.6.5) and uses simple SMPC protocols like *secure k-th element* for distributing the split operations of Mondrian. This protocol does not suffer from the information leakage we identified in this work at the cost of only allowing binary splits of numerical or ordinal data in comparison to arbitrary taxonomy trees. Furthermore, the authors introduce *l-site-diversity* as a special syntactic privacy models for the distributed setting. Tassa and Gudes [TG12] provide algorithms for the anonymization of horizontally and vertically partitioned data supporting k-anonymity, l-diversity and l-site-diversity. The algorithms are based on sequential clustering and employ simple SMPC algorithms for secure sums and boolean AND operations. Goryczka, Xiong, and Fung [GXF14] introduce the notion of *m-privacy* dealing with colluding parties in a distributed anonymization scenario and provide algorithms for the anonymization of horizontally partitioned data with respect to k-anonymity or l-diversity (see Sections 2.6.2 and 2.6.3). The algorithm is based on the idea of Mondrian and simple SMPC protocols employing secret sharing. Kohlmayer et al. [Koh+14] introduce a flexible framework for anonymizing horizontally and vertically partitioned data. It is based on deterministic and commutative encryption and supports different syntactic privacy models. Chen et al. [Che+15] provide a differentially private version of the protocol of Mohammed et al. [Moh+10] which outputs noisy counts of equivalence class records after a fixed number of specializations and relies on a semi-trusted data aggregator and crypto service provider.

5.1.2 Secure Multi-Party Computation

SMPC is a cryptographic technique that allows multiple parties to jointly compute the output of a public function $f()$ on their private inputs, without revealing any information about these inputs to each other, apart from what can obviously be deduced from the functions output. The classical example in SMPC is Yao's millionaire's problem, where a certain number of people want to determine who is the most wealthy among them, without revealing their actual wealth (for example, their account balances) to each other.

In some sense SMPC allows computation on encrypted or masked data, offering various degrees of security, according to the requirements of the specific deployment scenario.

SMPC Protocols

There are several SMPC protocols that enable secure and privacy-preserving computation, whose origins date back to the 1980s. The two-party SMPC protocol Yao's garbled circuits [Yao86] is based on an encrypted ("garbled") form of a Boolean circuit, which represents the function $f()$ that the two parties want to compute. One party (called the garbler G) creates this garbled circuit and securely encodes their own inputs into it. The other party (the evaluator E) gets a garbled version of their inputs privately via OT [Ish+03]. E also receives the garbled circuit, which E can then evaluate using their encoded inputs in order to reveal an encoded representation of the function's output. G and E can then jointly decode the plaintext output. Yao's protocol was extended to more than two parties in the *BMR* protocol [BMR90].

Another protocol that is commonly used in SMPC is the *GMW* protocol [GMW87], which uses exclusive OR (XOR)-based secret-sharing to mask private values and process them in a Boolean Circuit. The core idea in GMW is to chose random masks (called "shares") for every value such that the XOR of all shares adds up to the plaintext value. These shares are distributed to all parties that can then jointly run the computation. Certain operations (namely data-dependent AND gates) require interaction and the computation of OTs between the parties in order to correctly update each shares after the operation. The circuit's output is a secret-shared value that the parties can reconstruct by exchanging their shares. The GMW protocol can naturally scale to an arbitrary number of parties ≥ 2. It can also easily be modified to handle arithmetic values with bit-length k in the ring $\mathbb{Z}_{2^k}$ by using additive secret-sharing.

There exist further protocols like *SPDZ* [Dam+12] which is also secret-sharing-based, allows computation for 2 or more parties, but originally works in prime fields.

SMPC Frameworks.

The first implementation of an SMPC protocol was *Fairplay* [Mal+04] that brought the theoretic constructs of the 1980s into the realm of practicality. *ABY* [DSZ15] was a framework for secure 2-party computation with passive security and the ability to securely convert between the above-mentioned protocols. *MOTION* [Bra+22] extends ABY's functionality to ≥ 2 parties and further optimizes several protocol aspects. Implementations of *SPDZ* and other protocols, providing several levels of security, are implemented in *SCALE-MAMBA* [Aly+22] and *MP-SPDZ* [Kel20].

In the past few years, a vast body of research emerged that also covers various levels of security and certain specific numbers of parties. The authors of MOTION [Bra+22] provide a broad overview of those protocols and their corresponding implementations.

5.2 A Vulnerable Distributed Syntactic Privacy Protocol

In this section we present the protocol of Mohammed et al. [Moh+10]. It extends the ideas from [FWP07] to the distributed setting with horizontally partitioned data. For reasons of better transferability, we mainly stick to the notation of Mohammed et al. in the following sections.

In the distributed anonymization setting for horizontally partitioned data n parties own disjoint local data tables $T^{(i)}$. Each record in these tables has d attributes from domains $\{D_1, \dots, D_d\}$ and an additional sensitive attribute, such as the disease a patient is suffering from, from the domain of sensitive values $Sens$. Some of the attributes are defined as QIDs which in combination are suited to link individual records and therefore their sensitive value to individuals. The goal of the distributed syntactic privacy protocol is to determine a dataset T' from the combined dataset $T = \cup_{i=1}^{n} T^{(i)}$ which satisfies the targeted privacy model. This should be performed in a way that does not allow parties to learn more information than the final table T' offers. This informally phrased privacy requirement is covered in more detail at the end of this section.

The protocol allows the distributed computation of an anonymized dataset fulfilling a k-anonymity-based syntactic privacy model called LKC-privacy. LKC-privacy tries to deal with the *curse of high dimensionality* (see Section 2.6.7) – the effect that a larger number of attributes leads to high information loss in k-anonymity due to a sparse distribution of data points in the high-dimensional space. The general idea of LKC-privacy is to assume bounded adversary knowledge in the number of known QID attributes given by parameter L. An anonymized dataset fulfills LKC-privacy iff any combination of at most L QID attributes occurs at least K times. Additionally, to prevent *homogeneity attacks* (see Section 2.6.8), the ratio of any sensitive value in any group of records sharing the same QID attributes with respect to L must not surpass C. However, in our opinion, the assumption of limited attacker knowledge in the number of QID attributes is an improper weakening of the privacy guarantees of k-anonymity. In practice, there exist scenarios in which it makes sense to limit attacker knowledge based on different data sources. For example, a doctor in a hospital has access to different data attributes than an employee of an insurance company. Limiting only the number of possibly known attributes, on the other hand, is an oversimplification. For this reason, we cover a variant of the protocol which results in a k-anonymous dataset instead of one fulfilling LKC-privacy. This is achieved by a minor change to the way the validity of further protocol steps is determined. The exchanged data remains the same and the information leakage described in Section 5.2.3 is not affected in any way.

The authors provide two protocol variants which target two different use cases: classification and general data analysis. While these protocols differ slightly, the information leakage exists in both variants. Therefore we describe the protocol for general data analysis because of its wider applicability. In the following we cover the relevant protocol steps in detail and point out where the protocol would deviate for LKC-privacy or the classification task.

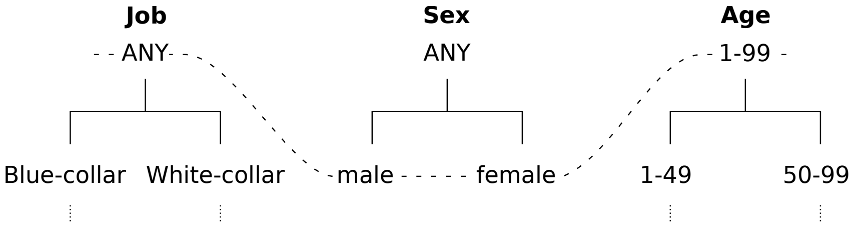


Figure 5.1: Taxonomy trees and $Cut = \{ANY Job, male, female, 1-99\}$. Figure based on [Moh+10].

5.2.1 Protocol Description

Recall that the protocol involves n parties which each have a set of data records with the same attributes, since the data is horizontally partitioned among the parties. The parties are assumed to communicate in a ring topology. One of the parties takes the role of the leader being responsible for organizing the anonymization process. The basic protocol idea is to anonymize a table by specializing QID attributes one at a time as long as there are valid specializations, starting with the most general value for each QID attribute D_i (which is also known as a *top-down* strategy [JC06]).

For each QID attribute the protocol expects a *taxonomy tree*[1] for possible specializations. Figure 5.1 provides example trees for different QIDs. Leaf nodes of such trees describe domain values and parent nodes more general values. A specialization $v \rightarrow child(v)$ describes the replacement of a specific QID attribute value v with the appropriate child value $c_j \in child(v)$, where $child(v)$ describes the set of less generalized values of v. An example based on Figure 5.1 is the specialization $ANY \rightarrow \{Blue{-}collar, White{-}collar\}$ of the job attribute. In the protocol of Mohammed et al. specializations are performed for all records containing v simultaneously which is also known as *global recoding* (see Section 2.5.1). For the taxonomy tree of QID attribute D_l we keep track of the current state of specialization by looking at the cut Cut_l of the tree which contains one value on each root-to-leaf path of the tree given by performed specializations. The union of all cuts $Cut = \cup_{l=1}^{d} Cut_l$ determines the current state of specialization for all QID attributes. Performing a specialization $v \rightarrow child(v)$ can be interpreted as pushing the cut down for a specific QID attribute value v by replacing v with $child(v) = \{c_1, \ldots, c_j\}$. An example for a possible cut after one specialization is shown in Figure 5.1. The protocol starts by initializing Cut with the topmost values of all taxonomy trees for QID attributes $D_1, \ldots, D_d$ – a state in which no specialization has been applied.

In addition, the current state of specialization given by the current Cut determines a set of current *equivalence groups*. An equivalence group (in other publications also referred to as *equivalence class* [LLV07]) describes the group of records sharing the same values for all QID attributes. For example, in the first round of the protocol we have $Cut = \{ANY Job, ANY Sex, 1-99\}$ and a single resulting equivalence group $E_{ANY Job, Any Sex, 1-99}$.

As the first protocol step, the leader collects initial *count statistics* cs. These are gathered via *Information* messages sent from party to party via the communication ring. Count

[1] We refer to these taxonomy trees as *generalization hierarchies* (see Section 2.5.1), but stick to the notation and wording of Mohammed et al. [Moh+10] in this chapter.

statistics provide a global view on the distribution of all parties' data records with respect to the current equivalence groups as determined by the current Cut. For each of these groups the count statistics cs contain the number of records in this group. Additionally for each QID attribute value v_l in an equivalence group the count statistics cs contain the number of records in all child equivalence groups if the specialization $v_l \rightarrow \{c_1, \dots, c_m\}$ would be performed[2]. The authors provide an example for count statistics cs for the initial equivalence group $E_{ANY\,Job, ANY\,Sex, 1-99}$:

```
(ANY Job, ANY Sex, 1-99, 3):
      (ANY Job, 2, 1),
      (ANY Sex, 2, 1),
      (1-99, 3, 0)
```

This is to be understood as follows: The depicted equivalence group contains three records. If the QID attribute *sex* is specialized, the resulting equivalence groups $E_{ANY\,Job, male, 1-99}$ and $E_{ANY\,Job, female, 1-99}$ will contain two respectively one records.

To prevent the parties from learning individual contributions to the global count statistics, a simple secure sum protocol [Sch07] is employed for this step and all subsequent computations of count statistics. Let p_i be the record count of party P_i for a specific equivalence group E_v or possible child equivalence group E_{c_j}. The goal is to compute the sum $\sum_{i=1}^{n} p_i$ over all record counts. The leader chooses a random integer $r \in \mathbb{Z}$, adds its record count p_1 and sends the result to P_1. Each party P_i adds their input p_i to the received sum and sends the result to P_{i+1}. Finally P_n sends the result $r + \sum_{i=1}^{n} p_i$ to the leader who can compute the final sum by subtracting r.

The count statistics cs are used by the leader to determine the validity of possible specializations in the current Cut. A candidate specialization $v \rightarrow child(v)$ for $v \in Cut$ is valid if the table resulting from this specialization does not violate k-anonymity. This means that for every $c_j \in child(v)$, all equivalence groups E_{c_j} contain at least k records[3]. Furthermore, the leader also uses the count statistics to find the best specialization $Best \rightarrow child(Best)$ among all valid specializations using the *discernibility cost* as a scoring function[4]. This scoring function is computed as $Score(v) = \sum_{E_v} |E_v|^2$ and favors specializations of larger equivalence classes. The best candidate for specialization is given by the specialization with the highest score.

Based on this information the leader chooses the valid specialization with the highest score and informs the other parties about this next specialization $Best \rightarrow child(Best)$ via an *Instruction* message. Each party performs this specialization on their local data. The use of a suitable data structure called *Taxonomy Indexed Partitions* (TIPS) reduces the necessary computations for the new local data record counts (details can be found in section 4.2 of [Moh+10]). Afterwards updated count statistics cs are collected

[2] For classification analysis additional record counts for the distribution of classification labels would be contained in the *Information* messages.

[3] This validity criterion differs for LKC-privacy. Using the same count statistics cs, we determine if for each $c_j \in child(v)$, every set qid_{c_j} of QID attributes of size L containing c_j leads to equivalence groups $E_{qid_{c_j}}$ with at least K records and no sensitive attribute $sens$ appears in these groups with probability larger than C.

[4] For classification analysis, the score would additionally depend on the classification label distribution to choose the specialization with the highest information gain.

to compute the validity for all specializations possible in the next iteration of the protocol. These steps are repeated until no more valid specializations are found. Then the local dataset $T^{(i)}$ is transformed to $T'^{(i)}$ by each party according to the performed specializations. Finally the resulting datasets are integrated into a final dataset $T' = \sum_{i=1}^{n} T'^{(i)}$ fulfilling k-anonymity by using a *secure set union protocol* [JX08] which hides the data origin from other parties.

5.2.2 Privacy and Security Claims

Mohamed et al. [Moh+10] assume the semi-honest (also referred to as *passive* or *honest-but-curious*) adversary model [PMB14], in which adversaries do not deviate from a given protocol but try to learn as much information as possible from messages legitimately received during the protocol execution. They claim that parties in the protocol do not share more detailed information than what can be extracted from the final integrated table T' and especially that exchanged information in the count statistics cs do not violate this requirement. In particular, the child equivalence group record counts determining if an equivalence group E can be further specialized do not reveal anything more than the final integrated table because “a specialization should take place as long as it is valid”. The authors mention that the integrated data is less anonymous to data holders as they can always remove their own data from the anonymized table T' which possibly results in a violation of k-anonymity. Finally, they explicitly state that the party acting as leader during the protocol does not need to be more trustworthy than others.

5.2.3 Weaknesses in the Protocol

According to the privacy and security claims no party should learn more about other parties and their data than what the final generalized, integrated table T' reveals. Parties can only deduce further information by removing their own data records from the final table. However, we identified another information leakage which allows the leading party to deduce more information about individual records than the final table T' allows (completely independent of local data). This information leakage emerges from the final count statistics the leading party collects for an equivalence group E which cannot be specialized further in the next step without violating k-anonymity. Because no further specialization is possible the count statistics for the equivalence group must contain record counts for child equivalence groups E_{c_j} being smaller than k. This leaks additional information about the distribution of attributes in the equivalence group E and in particular more information than can be deduced from the final table T'.

In the following we provide a small example in the medical domain highlighting the consequences which might follow from this additional information obtained by the leading party. We use our well-known QIDs job, sex and age and the patient’s disease as sensitive attribute. Imagine the following as a part of the final count statistics cs:

```
(White-collar, ANY Sex, 50-99, 5):
      (White-collar, 3, 2),
      (ANY Sex, 4, 1),
        (50-99, 2, 3)
```

Obviously any further specialization would violate the requirement of at least $k = 5$ data records in all resulting equivalence groups. For example, specializing the age attribute 50-99 would lead to equivalence groups with two and three data records. Further, imagine the relevant part of the final anonymized result table T' for the given equivalence group $E_{White-collar,ANY sex,50-99}$ gathered by the leading party as given below.

Job	Sex	Age	Disease
White-collar	ANY	$50-99$	Testicle cancer
White-collar	ANY	$50-99$	Breast cancer
White-collar	ANY	$50-99$	HIV
White-collar	ANY	$50-99$	Testicle cancer
White-collar	ANY	$50-99$	Influenza

By employing background knowledge about the prevalence of breast and testicle cancer in the male and female population, we can conclude that there are at least two male patients and at least one female patient present with high probability. Furthermore we know from the final count statistics that specializing the equivalence group by sex would lead to equivalence groups with one and four data records. By combining these information we can infer that there is a single woman between 50 and 99 years of age employed in a white-collar job in our dataset who suffers from breast cancer. This inference would have not been possible from the final table T' without the additional information leaked by the final count statistics. Note that, depending on the data distribution, similar conclusions might be drawn for all possible further specializations and final equivalence groups.

A further weakness of the protocol with respect to the privacy claims arises from a well-known weakness in the used *secure sum* protocol [Cli+02]. Colluding parties P_{i-1} and P_{i+1} are able to reconstruct the record counts of P_i by subtracting the counts P_{i+1} received from the ones P_{i-1} sent. Information about the local data distribution of parties might even be possible for a set of victim parties being surrounded by adversarial parties, for example the maximal number of records in equivalence classes or their absence. Note that relying on the semi-honest attacker model does not eliminate this attack since it involves the possibility of colluding parties [EKR18].

5.2.4 Adversary Model of the Original Protocol

Based on this analysis we provide the realistic adversary model for the original protocol. As already stated, Mohammed et al. assume the semi-honest adversary model, in which adversaries do not deviate from the protocol but try to learn as much information as possible from legitimately received protocol messages. In our opinion, this is a valid assumption since the scenario inherently dictates trust being placed in the data-providing parties to some extent (for example by requiring them to perform computations on their real local data) for a correct execution of the protocol.

Even though not explicitly stated by Mohammed et al., communication connections around the ring are assumed to be secure so that no outside adversary nor any other party can access or tamper with the message content. Otherwise an adversary might be able to obtain equivalence class counts of individual parties similar to the ones obtained through the weakness in the secure sum protocol.

The leading party has to be more trustworthy than other parties due to the described weaknesses in the protocol. As we have shown, an adversarial leading party can break the k-anonymity guarantees. Parties are not allowed to collaborate in an adversarial manner, that is beyond communication that is required by the protocol. Otherwise they might be able to deduce specific equivalence class counts of other parties (depending on their position in the communication ring) due to the weakness in the employed secure sum protocol.

A final note: As Mohammed et al. [Moh+10] state, parties are more trusted in that they can always remove their own data from the final table. There are two potential solutions to this problem: either this is assumed acceptable in the application scenario or the parties must not get access to the final table. In this case an additional party without any data themselves can act as the leading party and must be the only party which obtains the final k-anonymous table. This is possible due to the employed secure set union protocol in the final protocol step.

5.3 Improving the Protocol through Secure Multi-Party Computation

In this section we present our approach to prevent the weaknesses in the protocol by Mohammed et al. [Moh+10]. We cover the basic idea of our approach, its implementation based on a specific SMPC framework, and finally the updated adversary model of our solution.

5.3.1 Basic Idea

First, we just consider a single equivalence group E_v and the specialization $v \to \{c_1, \ldots, c_m\}$ of this equivalence group with respect to a single quasi-identifier, for example, the specialization of age value $v = ANY$ to age groups $c_1 = 0 - 49$ and $c_2 = 50 - 99$.

The main idea behind our approach is based on the fact that assessing the *validity* of the specialization does not require the exact record counts $|E_{c_1}|, \ldots, |E_{c_m}|$ of the resulting child equivalence groups $E_{c_1}, \ldots, E_{c_m}$. It is sufficient to know whether each child equivalence group contains more than k records (or none which also does not put individual data records at risk). The real counts are just required for the score computation when choosing the next best specialization. Since invalid specializations must not be performed, record counts for the child equivalence groups resulting from these specializations are not required at all. We can utilize this by using a method which just yields counts for child equivalence groups with none or more than k records and hides the counts otherwise.

In the original protocol a simple secure sum protocol is used which computes the sum $\sum_{i=1}^{n} |E_{c_j}|^{(i)}$ for child equivalence groups E_{c_j} based on the record counts of all parties $|E_{c_j}|^{(i)}$ $(i \in \{1, \dots, n\})$. Our goal is to achieve a protocol which can distinguish three cases: no records, at least one and less than k records, and at least k records, in which case the exact count should be provided. This allows us to hide the leaked information of the original protocol which occurs when child equivalence group record counts lower than k are received by the leading party.

Furthermore, we have to consider that whenever a child equivalence group E_{c_j} would contain less than k records, we have to hide the record counts of other child equivalence groups E_{c_x} for $x \neq j$ as well. Otherwise we could leak the hidden record count $|E_{c_j}|$ since the difference of parent count and the sum of other child equivalence group record counts $|E_v| - \sum_{x=1, x\neq j}^{m} |E_{c_x}|$ equals the hidden record count $|E_{c_j}|$. Hiding all child equivalence group record counts does no harm because the entire specialization is invalid as soon as a single child equivalence group would violate k-anonymity. In other words, we want to compute

$$f(E_{c_j}) = \begin{cases} k_{mask}, & \text{if there exists } c_x \in \{c_1, \dots, c_m\} \text{ with } 0 < \sum_{i=1}^{n} |E_{c_x}|^{(i)} < k \\ \sum_{i=1}^{n} |E_{c_j}|^{(i)}, & \text{otherwise} \end{cases}$$

for each child equivalence group Ec_j. The value k_{mask} describes an indicator for the fact that the sum of record counts is hidden.

The described functionality cannot be achieved by a protocol comparably simple to the one used in the original protocol. However we can use SMPC protocols like *GMW* (see Section 5.1) that allow to compute any finite function in a distributed manner without leaking the private inputs (in our case record counts) to the other parties.

5.3.2 Implementation via SMPC

In our updated protocol implementation we employ a SMPC framework called MOTION [Bra+22] which provides security against $n-1$ passive adversaries. Using the framework therefore directly prevents the attack of colluding parties which exists in the original *secure sum* protocol. Each protocol run in MOTION starts with an input-independent setup phase where required values are shared between all parties. Afterwards a provided circuit consisting of input gates, intermediary gates for boolean or arithmetic computations, and output gates is evaluated. MOTION offers the possibility to switch between arithmetic and boolean circuit based SMPC protocols. Since our functionality requires multiple additions as well as multiple comparison operations (see below), using arithmetic *and* boolean circuits and switching between them allows to combine the faster additions in arithmetic circuits with the faster comparison operations in boolean circuits. To achieve the desired functionality we have to provide it in the form of a circuit to MOTION. We will cover this circuit in two parts below.

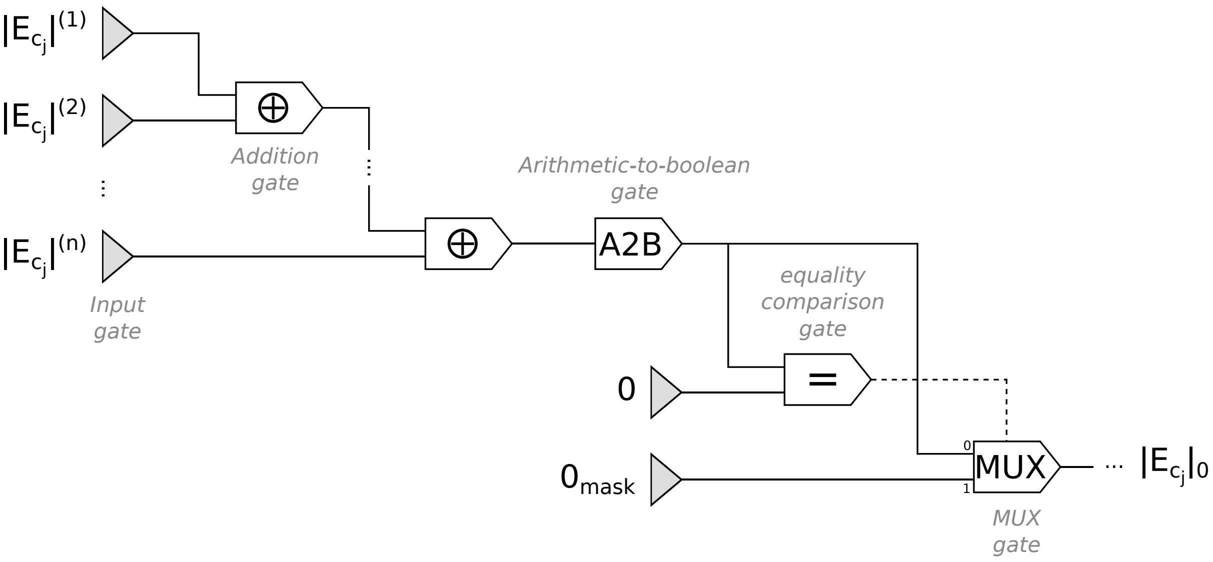


Figure 5.2: First part of the SMPC circuit used in our protocol. The depicted circuit part covers the counts for just one potential child equivalence class.

The first part of the circuit, visualized by Figure 5.2, computes the sum of all parties' (private) record counts $|E_{c_j}|^{(i)}$ for a specific child equivalence class and checks if this sum is larger than zero. For this purpose, the counts $|E_{c_j}|^{(i)}$ for child equivalence class E_{c_j} are provided to the circuit via *input gates* and afterwards added through *addition gates*. These computations are performed based on the arithmetic GMW protocol. Afterwards the resulting sum is converted to be usable in the boolean GMW protocol, since the remaining operations are comparisons or are dependent on comparison results. The next step is to compare the sum to zero in an *equal gate* to distinguish the case of empty child equivalence groups. We then use a *MUX gate* which outputs one of two values depending on the single signal bit being computed by the equality comparison. This gate outputs $|E_{c_j}|_0$ which is the real sum $|E_{c_j}|$ if it is larger than zero and a special mask value 0_{mask} otherwise. We use a integer for this mask value, which is larger than practical record count sums, so that we can compare the sum $|E_{c_j}|_0$ to k without the necessity to treat 0 as a special case. This circuit is performed for all child equivalence groups E_{c_j} in parallel ($j \in \{1, \ldots, m\}$).

The second part of the circuit (depicted in Figure 5.3) starts by comparing all sums $|E_{c_j}|_0$ to the anonymity parameter k via *larger-than gates*. We combine the resulting bits with *OR gates* to achieve a final bit which is 1 if any child equivalence group record counter $|E_{c_j}|_0$ is less than k, and 0 otherwise. This bit is used as a signaling bit for the final m *MUX gates* so that they output the sums $|E_{c_j}|$, if no child equivalence group contains less than k records, and another mask value k_{mask} for all child equivalence groups, otherwise. As a final result of this circuit we get m outputs for the specialization $v \to \{c_1, \ldots, c_m\}$ where each output is either the real sum $|E_{c_j}|$, a mask value 0_{mask} for empty child equivalence groups, or a mask value k_{mask} if there is at least one child equivalence group with less than k records.

This circuit – presented for one equivalence group E_v and one specialization $v \to \{c_1, \ldots, c_m\}$ – has to be performed for all equivalence groups and all possible specializations given by the Cut. But these computations can be performed in parallel so that we

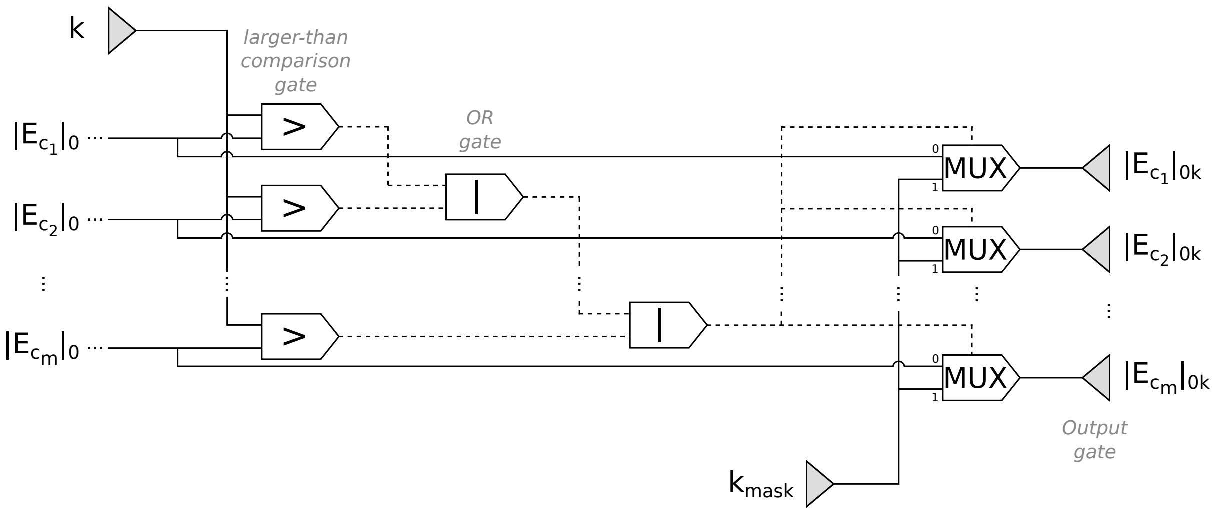


Figure 5.3: Second part of the SMPC circuit used in our protocol. This part shows the circuit for all potential child equivalence classes, that is, each input value $|E_{c_j}|_0$ has been computed by the circuit part depicted in Figure 5.2.

just need to compute a single MOTION circuit per round of the original anonymization protocol.

The described protocol can be used as a drop-in replacement for the simple *secure sum* protocol used by Mohammed et al. with the only difference that we require peer-to-peer communication compared to the ring topology used by the original protocol. The leakage in the protocol of Mohammed et al. we have identified in Section 5.2.3 emerged from record counts of potential child equivalence classes for specializations which would contain less than k records and therefore violate k-anonymity. Since these counts are hidden for all children in our updated protocol, we have effectively prevented the information leakage and potential privacy violations by the leading party. Our implementation of the original and updated protocol is publicly available[5].

5.3.3 Updated Adversary Model

The improvements of the protocol achieve a stronger attacker model in comparison to the one given in Section 5.2.4 for the original protocol. In the following we provide details about the adversary model for our improved protocol variant.

Our solution still assumes the semi-honest adversary model, which is a valid model as we have argued in Section 5.2.4. The utilized SMPC framework MOTION uses this assumption as well. In comparison to the ring communication in the original protocol, we additionally assume secure peer-to-peer connections between all parties, which are required by MOTION.

In contrast to the original protocol the leading party in our improved protocol does not require more trust than any other party. Because equivalence class records smaller than

[5] The repository can be found at `https://github.com/tompetersen/decentralized-syntactic-privacy`.

the threshold k are hidden from the leading party, the party cannot infer information violating the guarantees of k-anonymity.

The employed SMPC protocol MOTION provides *full-threshold* security, that is all but one party can be corrupted (in the sense of a semi-honest adversary) and the private inputs are still secure [Bra+22]. This property prevents the weaknesses arising from collaborating parties in the secure sum protocol in the original protocol.

Just like in the original protocol, in our variant parties can extract their own data from the resulting dataset in order to violate the k-anonymity guarantees. Potential solutions covered in Section 5.2.4 apply for the improved protocol as well.

5.4 Performance Evaluation

To compare the performance of the original protocol and our SMPC-based protocol and especially to identify the overhead that our solution imposes, we have conducted several experiments. The experiments were performed on a system with two Intel Xeon E5-2630 processors (10 cores, 2.20 GHz) and 128 GB of RAM. We have used a fixed anonymity parameter $k = 5$ for all experiments and employed two datasets:

- The ADULT dataset [Koh96] consists of 48 842 records of the 1994 US Census database. It includes amongst others the following attributes for each individual: the age, number of education years, marital status, occupation details, sex, and race.
- A non-public synthetic dataset from the medical domain consisting of 2 240 records just containing the age and sex of individuals as QIDs.

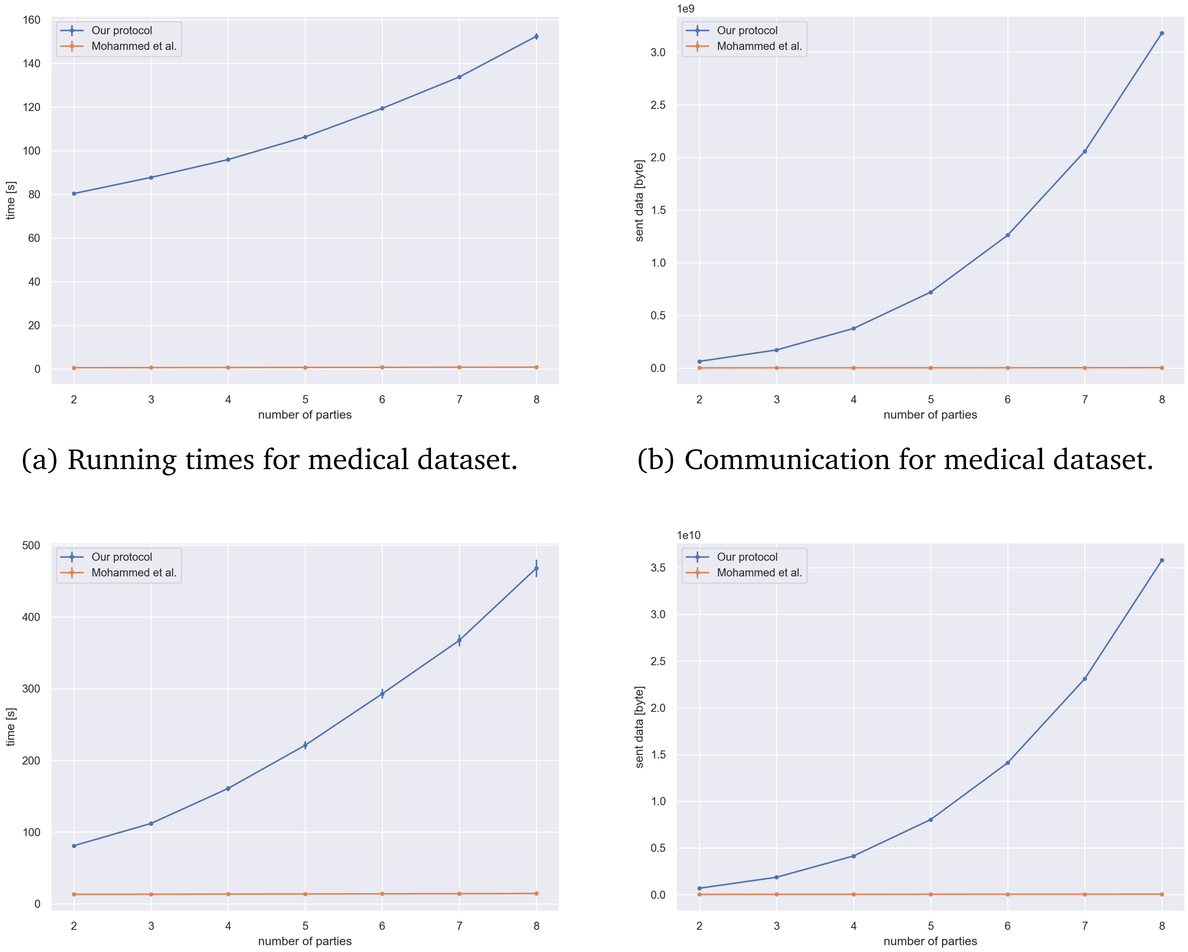
Our first experiments compare the runtime and communication demands of our protocol in comparison to the original protocol for different numbers of participating parties. During each experiment the respective dataset was partitioned into equally-sized parts according to the number of participating parties and all parties were simulated on the same system. The results for the medical dataset (using the two QIDs age and sex) and for the ADULT dataset (using the three QIDs age, gender, and occupation) are displayed in Figure 5.4. To better spot the overhead of our protocol, Table 5.1 additionally provides the exact running times and communication costs.

Our protocol imposes a heavy overhead in runtime and communication demands, which is not surprising as it is built upon a universal SMPC framework. This also explains the increase of communication overhead with each additional party, whereas the original protocol only requires constant additional resources for each party (given the simple ring structure used in the protocol).

Our next experiments determine the influence of using different sets of QIDs. For this purpose we performed our measurements with the ADULT dataset only, as it allows for more different QID combinations than our medical dataset, and used six QIDs (age, education years, marital status, occupation, race, and sex). We measured the runtime and communication demands for all possible combinations of these QIDs for a fixed set of three parties.

Table 5.1: Overall runtime and communication costs for the protocol by Mohammed et al. and our protocol for a varying number of parties. The given standard deviations for the runtime were collected across 10 runs. The amounts of exchanged data did not vary across runs, as expected.

		Runtime		Communication	
Dataset	Parties	Mohammed	Our Protocol	Mohammed	Our Protocol
Medical	2	0.6 s ± 0.1	80.4 s ± 0.3	1 MB	63 MB
	3	0.6 s ± 0.0	87.8 s ± 0.7	1 MB	171 MB
	4	0.7 s ± 0.0	95.9 s ± 1.0	2 MB	375 MB
	5	0.7 s ± 0.0	106.3 s ± 0.8	2 MB	720 MB
	6	0.7 s ± 0.0	119.4 s ± 0.6	2 MB	1.26 GB
	7	0.8 s ± 0.1	133.8 s ± 1.0	2 MB	2.06 GB
	8	0.8 s ± 0.0	152.4 s ± 1.5	3 MB	3.18 GB
ADULT	2	13.2 s ± 0.1	81.1 s ± 1.8	26 MB	687 MB
	3	13.4 s ± 0.1	112.1 s ± 3.2	30 MB	1.86 GB
	4	13.6 s ± 0.1	160.9 s ± 3.6	34 MB	4.14 GB
	5	13.8 s ± 0.3	221.1 s ± 5.8	38 MB	8.03 GB
	6	14.0 s ± 0.2	292.9 s ± 6.9	42 MB	14.12 GB
	7	14.2 s ± 0.1	367.3 s ± 8.3	46 MB	23.11 GB
	8	14.4 s ± 0.1	468.0 s ±12.2	51 MB	35.79 GB



(a) Running times for medical dataset.

(b) Communication for medical dataset.

(c) Running times for ADULT dataset.

(d) Communication for ADULT dataset.

Figure 5.4: Experimental runtime and communication requirements for medical and ADULT dataset depending on the number of parties. The error bars present the standard deviation for 10 runs.

The runtime displayed in Figure 5.5 is primarily dominated by the number of protocol rounds which depends on possible specializations and the number of performed specializations. The experiment with the lowest runtime performs the only possible specialization: specializing ANY sex to male and female. Using more attributes as QIDs in principle leads to higher running times. On the other hand, the *curse of high dimensionality* (see Section 2.6.7) also comes into effect: More QID attributes can cause less protocol rounds since fewer specializations already produce a state in which any further specialization would violate k-anonymity. This leads to lower running times as well. One example for this is the usage of *all* QIDs, which results in an average runtime compared to all experiments.

The communication complexity displayed in Figure 5.6 is not directly proportionate to the runtime. Even though the communication complexity depends on the number of rounds just like the runtime, the number of QIDs has a high impact as well. In each round the number of records in possible child equivalence groups has to be computed for the specialization of all QIDs. This causes larger circuits which require more communication for their evaluation.

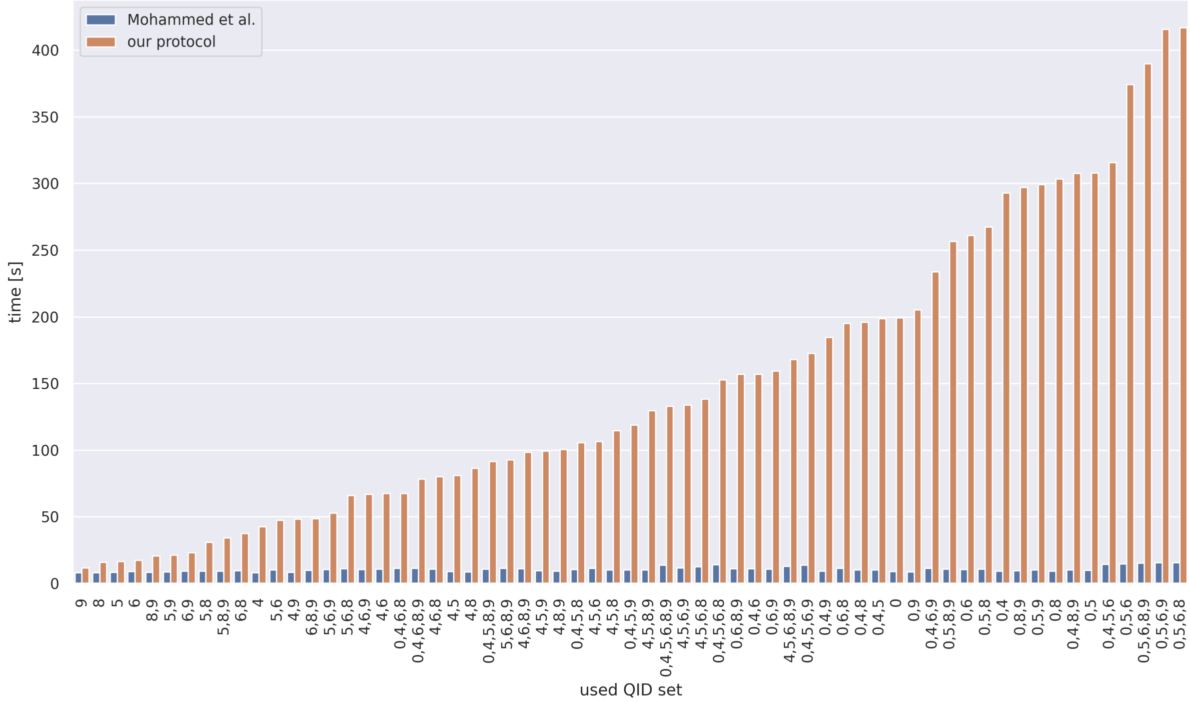


Figure 5.5: Overall running times for using combinations of the attributes age (0), education years (4), marital status (5), occupation (6), race (8), and sex (9) as QIDs (3 parties).

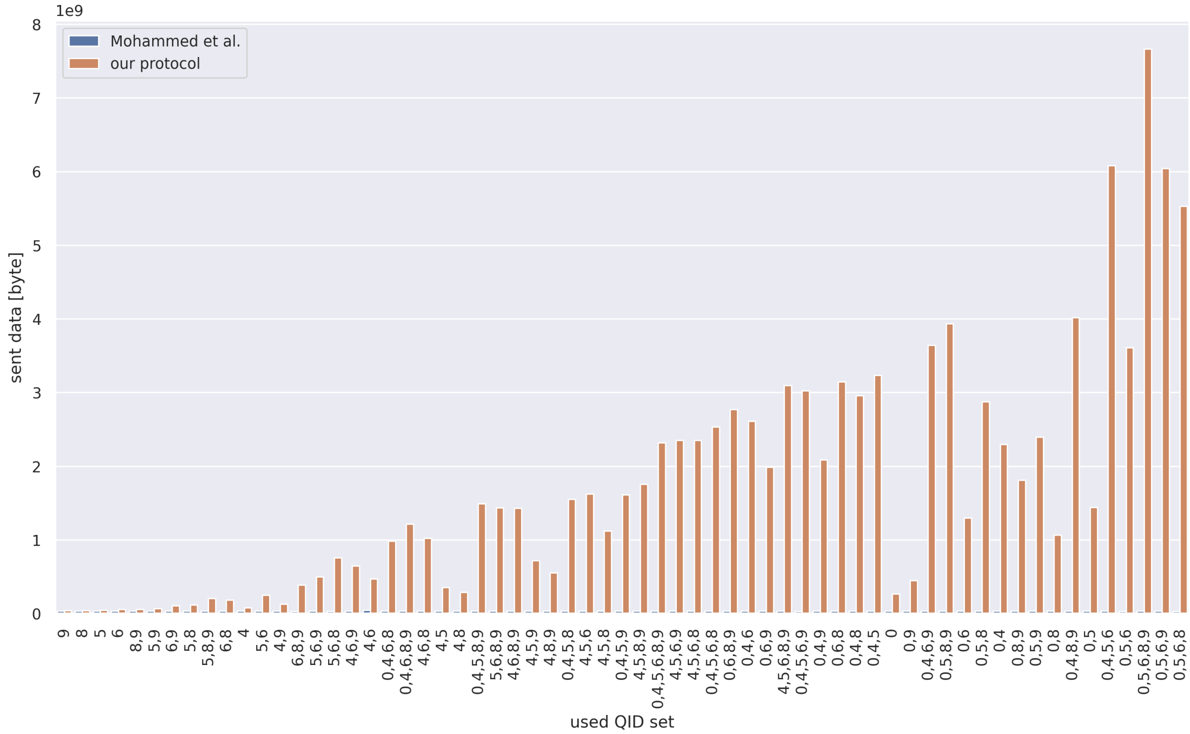


Figure 5.6: Overall communication demands for different QID sets ordered by increasing running times.

To summarize, using our updated SMPC-based protocol implicates an expected high performance overhead in terms of computation as well as communication demands. A simple first measure for reducing this overhead would be to pre-compute the data-independent values in the setup phase of the SMPC protocols just once and reuse these values across all protocol runs. This functionality is planned but unfortunately not yet implemented in MOTION[6].

5.5 Conclusion

In this chapter we have identified an information leak in a well-known distributed syntactic privacy protocol by Mohammed et al. [Moh+10]. The leak arises from the necessity in the protocol to collaboratively compute sizes of all potential next equivalence groups if a specialization takes place. The sizes of equivalence classes resulting from specializations that are not conducted due to k-anonymity constraints provide more insights into the attribute distribution within the equivalence class than the resulting k-anonymous table permits. A further weakness emerges from the simple secure sum protocol utilized by Mohammed et al. Colluding parties which are predecessor and successor of the victim party are able to compute the sensitive count statistics of the victim. We provided an updated protocol variant which prevents these weaknesses by replacing the simple secure sum protocol with a SMPC protocol which hides the size of a potential equivalence group E if $0 < |E| < k$. This variant prevents the information leak and additionally the vulnerability against colluding parties in the original protocol. The protocol was implemented based on the SMPC framework MOTION [Bra+22] and might be of independent interest in similar scenarios.

We have extensively evaluated the computation and communication overhead of the updated protocol. While this overhead is comparably high, it might be acceptable when dealing with sensitive (for example, medical) data – especially since publishing anonymized datasets should be performed only once for local datasets to prevent attacks (see Section 2.6.8). Furthermore, we observe an interesting consequence from the curse of high dimensionality in our evaluation: Running times and communication demands are related to the considered QIDs in a non-obvious way.

Directions for future research include the extension of our protocol variant to incorporate further syntactic privacy models like l-diversity (see Section 2.6.3) or t-closeness (see Section 2.6.4). These extensions potentially require further updates to the SMPC circuits to incorporate demands of the models' privacy conditions. Another line for improvements is to increase the performance via enhanced circuit design and the use of future SMPC framework functionality.

In the past chapters we have presented rather theoretical results regarding distributed pseudonymization and anonymization. The following Chapter 6 introduces a practical approach for distributed data collection in the medical domain. The method employs pseudonymization and anonymization techniques along with further privacy-preserving techniques.

[6] The respective issue can be found at `https://github.com/encryptogroup/MOTION/issues/4` (visited on 2024-09-13)

6 A Privacy-Preserving Medical Registry Platform

In healthcare research, *medical registries*, in addition to randomized controlled trials, play a crucial role in researching new therapies as well as improving and reviewing already established treatment procedures [Beh+23]. A medical registry is an organized system in which data about observation units, such as patients or medical devices, on a defined question is collected prospectively and in a standardized manner for a longer period of time [Nie+21]. These registries provide researchers with access to real-world patient data, enabling them to identify trends and evaluate treatment outcomes. However, concerns about patient privacy have led to increased scrutiny of how medical registries handle sensitive information [Beh+17b]. Ensuring the confidentiality of sensitive medical data while still enabling data analysis poses significant technical and organizational challenges. Traditional approaches to medical registry platforms often involve centralized databases where patient data is stored in plaintext and is accessible by a large user base.

In this chapter, we present a technical platform concept and implementation serving as a base for privacy-preserving medical registries. The platform supports the collection of data at multiple points in time, in other words, it allows for longitudinal studies. The main goal of the platform is the protection of patient's personal and medical data while allowing researchers to use the medical data for their research. For this purpose, we design a security architecture and chose appropriate security measures to achieve this goal. While legal considerations are essential for the legally compliant operation of medical registries, in this chapter we focus on technical aspects of such registries and omit legal details, such as the legal basis, consent management, or joint controllers.

Our main contributions are the following:

- We provide a concept for a medical registry platform with strong privacy and security guarantees which supports longitudinal studies.
- We enhance the platform with several features including monitoring and benchmark capabilities, ways for patients to execute data subject rights, and the option for patients to provide their own medical data.
- We propose an extensible data interface for researchers to export or query medical data in a privacy-preserving manner. The interface provides plugins employing syntactic privacy models as well as DP.

The chapter is structured as follows: In Section 6.1 we provide basic functional details of the platform including the general idea of the platform, participating roles, and features specific to the medical domain. Section 6.2 explains the security architecture of and specific measures taken in the platform. We provide details about the distinction between personal and medical data, the adversary model, the main security measure of cryptographically enforced client separation, as well as further measures. In the following two sections we cover further platform features in distinct sections. In Section 6.3 we describe the platform feature which allows patients to provide medical data to the platform themselves. Section 6.4 presents the data export and query interface for researchers, which offers privacy guarantees based on syntactic privacy models or DP through different plugins. Section 6.5 describes details about the platform implementation. In Section 6.6 we compare our platform to other open-source platforms for medical registries and Section 6.7 concludes the chapter.

The general platform concept presented in this chapter has been published [Pet+19]. More information about the concept and especially further details about regulatory requirements were described in the (non-public) data protection concept [Beh+21]. The developed platform was used in two medical studies. The *IDOMENEO* study [Beh+17a] examined the reality of care for patients suffering from PAD in more than 30 medical centers in Germany. As part of the study, data from more than 5,600 patients was collected in our platform [BD21]. The ongoing *INCREASE* study [Klo+22] investigates the use of modern therapy concepts in minimally invasive heart valve procedures. The mobile app described in Section 6.3 was implemented by Krause [Kra20] during his master's thesis. Preparatory work for the data interface's DP plugin was performed by Krass [Kra23] in his master's thesis in the form of a preliminary review of existing DP frameworks and some initial implementation attempts.

6.1 Functional Description

The basic idea of the platform is to allow a number of medical centers, such as hospitals, to record medical research data about their patients, to collect this data in a central place, and to provide researchers with the possibility to use this data in their research. For this purpose there is a central software component as well as distributed software components in all participating medical centers. Figure 6.1 shows a high-level overview of the platform. In the following, we provide an overview of its functionality.

The platform provides the so-called *survey admin*, the person responsible for organizing the medical study, with the possibility to create new medical centers and respective medical center admin accounts. In medical centers patients are examined and treated. Data is collected during these procedures by the medical staff and they are tasked with entering the patient's data. Each user of the platform in a medical center can take on one of three roles (*study nurse*, *doctor*, or *medical center admin*) with different responsibilities and rights. Details about the different roles in the platform are given in Section 6.1.1.

For each patient *personal data*, such as their name, address, or social security number, as well as *medical data*, such as measurements of vital functions, drug doses, or treatment success, are stored (see Section 6.2.1 for an in-depth discussion of this distinction). The personal data is used for data quality, documentation, and contact purposes. The medical data forms the basis for research purposes, such as studying the relationship between risk factors like smoking behavior and treatment outcomes. The platform provides forms to enter personal and medical data. These forms consist of different fields. The platform supports fields for different data types as well as data-type-dependent validation for individual fields and logical relationships between fields. There can be as many different forms and sub-forms as required by the scenario. To document the current review state

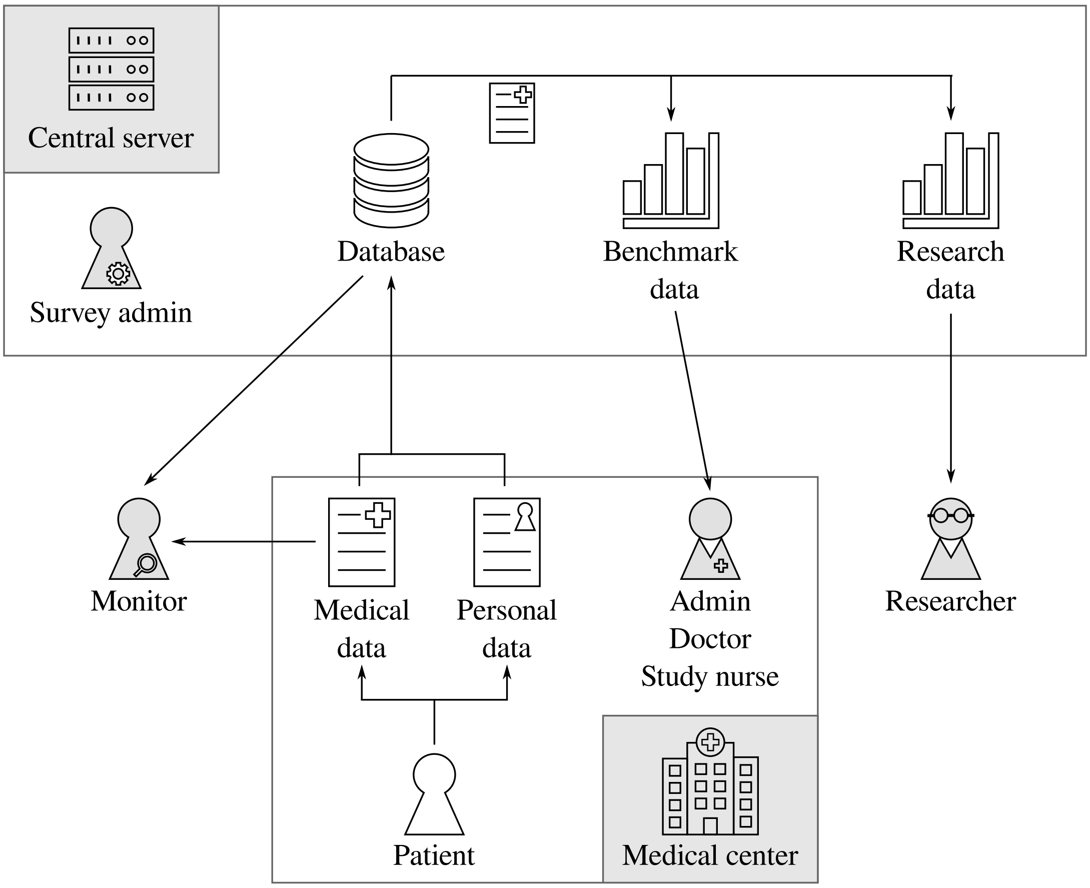


Figure 6.1: Platform overview for an individual medical center including data flow and user roles.

of form data, forms support multiple states from incomplete to fully entered and checked. This functionality is covered in Section 6.1.2. To ensure the correctness of entered data, external reviewers (*monitors*) can visit medical centers and compare the data entered in the platform with data from physical or digital health records present at the medical center. Details are given in Section 6.1.3.

A benefit for participating medical centers is the benchmark functionality provided by the platform. The platform makes data available to medical centers which allows them to compare their own patient-centered care to others, so that they can potentially learn from other centers. The benchmarking functionality is covered in Section 6.1.4.

The GDPR grants data subjects several rights in relation to the processing of their data. The platform allows patients to execute these rights. Details can be found in Section 6.1.5.

There are two other functions we cover in separate sections due to their extent and because they require details about security measures given in Section 6.2.

- Some medical studies are based, among other things, on data that describes a patient's health status on a daily basis. Often this data can be collected by the patient themselves. The platform allows patients to transfer this data digitally, as described in Section 6.3.
- Finally, the platform provides data export and query functionality for researchers while preserving patient's privacy. Section 6.4 provides more details about this functionality.

6.1.1 Role Model

The platform provides a tiered role concept with differing access rights for different roles to follow the *need-to-know principle*:

- **Study nurses** enter personal and medical patient data in the system. This data can, for example, originate in paper-based health records or in patient-filled forms.
- **Doctors**, just like study nurses, have the right to enter data. Additionally, they can accept completed data records after thorough review (see Section 6.1.2).
- **Medical center admins** are responsible for a single medical center. They create accounts for doctors and study nurses. Additionally, they can perform the tasks of study nurses or doctors.
- **Monitors** are responsible for reviewing the validity of data created in medical centers. For this purpose, after a successful request to the medical center admin, they get time-restricted access to medical data and paper-based access to personal data locally in the medical center.
- **Researchers** can apply for full medical data access or the results of specific computations on the data. Survey admins or an external committee can the accept or reject this application.

- **Survey admins** are responsible for organizing and running the study. They create new medical centers and pass initial account credentials to the responsible medical center admin. Furthermore, they create accounts for researchers and monitors. Depending on the scenario, they can access the combined medical research data of all medical centers, but never any identifying personal data.

There are further roles which are not part of the medical domain but are required for the operation of the platform:

- **Developers** write the code the platform functionality is based on. They should not get access to any data (medical or personal).

- **Server admins** operate the hardware the platform is executed on. Just like developers, they should not get access to any data (medical or personal).

6.1.2 States of Data Records

Medical data records offer different states: *draft*, *complete*, *accepted*, and *monitored*. These states describe different levels of validity of the data. The draft state is used for incomplete data records or data records likely to change and is automatically set for new data records. After entering and checking all data, the study nurse sets the state to complete. To enforce a multi-eye principle regarding data validity, a doctor is required to review the entered data and set the accepted state for the data record. If the data record is chosen for monitoring (see Section 6.1.3), the monitor checks the data again and finally sets the monitored state – the highest level of validity present.

6.1.3 Monitoring

Monitoring in this context means the process of verifying the correctness of collected data by an entity not part of the medical center. The responsible users, hereinafter referred to as monitors, compare the medical data entered in the platform to the data present in the medical center, for example, based on electronic health record (EHR). The personal data is not monitored since this process is just supposed to guarantee the validity of the medical data and scientific questions answered based on this data. To follow the principle of data minimization and to not tamper with the encryption of personal data solely for the medical center users (see Section 6.2.3), the monitors only get access to the medical data. The connection between pseudonyms and real patient identities (see Section 6.2.3) is provided to them in paper form at the medical center on site, where they also have access to treatment documentation. They mark medical data records as *monitored* or create queries for erroneous data. Users of the medical center can then correct these errors afterwards.

6.1.4 Benchmarking

Benchmarking in the medical domain describes performance and result comparisons between different healthcare providers with the aim of learning from each other through the structured exchange of experiences [KGS11]. In the platform this functionality is provided through statistics accessible by all medical centers. These statistics might include patient's demographic data, risk factors, or treatment details and success, aggregated for each medical center respectively. All users of a medical center are allowed to access these statistics of all medical centers to assess their treatment success in comparison to others. This may lead to insights calling for action, for example when a hospital experiences notably more treatment-related side effects compared to other hospitals treating patients with similar demographic characteristics.

Privacy considerations play an important role in this concept. Publishing even aggregate statistics can negatively impact the privacy of patients, for example, for medical centers with small patient numbers or for outliers with respect to demographic data. A first data-independent measure the platform takes is to use pseudonymous medical center names and to change their order in the published order randomly (but consistently over all published statistics). But care has to be taken when choosing the statistics to be published. In the spirit of the data protection principle of data minimization, one should pay attention to publish only statistics relevant for the given context. Additionally, further measures like generalization or suppression approaches (see Sections 2.5.1 and 2.5.2) should be considered. For example, this might include publishing only generalized age distributions or removing outliers.

The publication of regularly updated statistics for updated collected data requires particular attention. In these scenarios the pseudonymization and random ordering of medical center statistics might be easily reversed, for example, in the presence of outliers or by tracking statistical distributions. Differences between consecutive data distributions might allow deductions about patients being treated in the timespan between the publication of these distributions. This problem is similar to the weaknesses which can occur when releasing multiple syntactically anonymized datasets (covered in Section 2.6.8).

A possible solution, not implemented at the moment of writing, would be to employ DP for the computation of these statistics. This would require carefully choosing the total privacy budget ε and deciding on how to split this budget over multiple statistics and potentially over multiple timespans. Ideas and caveats can be taken from considerations about our privacy-preserving data interface (see Section 6.4), even though the context of the benchmarking functionality differs from this interface in its purpose, the trust placed in the user, and in the per-medical center requirement of the benchmarking.

6.1.5 Offering Data Subject Rights

The GDPR grants data subjects a number of data subject rights in accordance with articles 12 to 23 including amongst others the right of access, right to rectification, and right to erasure. The platform enables patients to execute these rights. The central platform component has no access to personal data since it is encrypted in a way that only medical centers can decrypt and access the data (see Section 6.2.3). Therefore the component would not be able to act on the request of a patient to execute their rights since it cannot even determine the correct data records belonging to a specific patient. For this reason data subject rights must be performed by the medical center which has treated the patient and collected their data.

6.2 Security Architecture and Measures

In this section we provide details about security measures implemented in the platform to protect patients' privacy. In Section 6.2.1 we provide preliminary details about collected data and the separation into medical and personal data. Section 6.2.2 establishes the attacker model for the platform as a basis for the following sections. The central security measure of the platform is the separation of personal patient data collected at different medical centers based on cryptographic techniques and pseudonymization introduced in Section 6.2.3. Sections 6.2.4 and 6.2.5 provide details about password security in the platform – an important aspect since these passwords are used for authentication as well as key derivation. But additionally to security requirements the safety of cryptographic keys plays an important role as well, since lost keys can potentially render previous efforts to collect study data useless. Measures for key safety are covered in Section 6.2.6. Another security measure concerns data access by developers while performing maintenance tasks. To prevent any unauthorized data access during these tasks, a maintenance mode has been implemented. This aspect is discussed in detail in Section 6.2.7. Finally, in Section 6.2.8 we introduce measures to impede the re-identification of patients through the pseudonymized medical data.

6.2.1 Personal and Medical Data

As already mentioned, we differentiate between patient's *personal data* and *medical data*[1]. *Personal data* refers to data which can be used to identify a patient, such as health insurance identifier, name, address, or the date of birth. *Medical data* is describing facts about a patient's health or medical issues, for example, treatment documentation or the patient-reported quality of life. This distinction is not necessarily exclusive. For example, the patient's age can play a vital role in medical decisions about the right treatment but can also be used for identifying purposes. It is also related to the classification of attributes in syntactic privacy models in directly identifying attributes, QID, and (non-)sensitive attributes (see Section 2.6.1). As covered in Section 2.6.1, in the domain of syntactic anonymization there are advocates for treating all attributes as potentially identifying. Therefore, the distinct classification used in our platform is artificial to some extent. Another take on this distinction is to classify medical data as data not directly identifying a patient, but fulfilling a medical research purpose.

In our platform, the personal data is stored in a way that allows only the patient's medical center to access this data (see Section 6.2.3). The medical data, on the other hand, can be accessed by the platform's central component for purposes such as research data export (see Section 6.4) or benchmarking (see Section 6.1.4). But it is still processed securely to prevent any unnecessary or adversarial access.

[1] In the German-speaking medical domain these concepts are often referred to as *IDAT* and *MDAT* [For21].

One might ask why the processing of personal data is required at all. There are several reasons for this. First, our platform aims at providing support for longitudinal medical studies, that is, studies in which medical data related to a patient is collected at multiple points in time, for example, after the initial treatment and during follow-up examinations. The medical staff has to be able to connect acquired data with the correct existing data record of the patient by means of a pseudonym (see Section 6.2.3). Additionally this connection is also required for the execution of data subject rights (see Section 6.1.5). Since our platform handles pseudonymized (and not anonymized) data, the GDPR still grants patients several data subject rights. But to execute these rights, the correct data records for a patient must be detectable. A second reason is that under certain conditions it can be necessary to re-identify a patient. Examples include data quality management through monitoring (see Section 6.1.3), special research interest in case of extraordinary disease conditions, or the delivery of warnings about discovered drug side-effects. Finally, the personal data is also used for convenience functions such as the automatic generation of personalized patient consent documents.

A possibility would be to not use the platform for storing the personal data, but to rely on some distinct solution for mapping the patient's identity to the given pseudonym – local to the medical center or via a TTP. There are several potential approaches for this.

- A first idea would be to store the mapping in the medical center by means of some platform-unrelated software or even in not-digital paper format. But this would not only impede the usability for users of the platform, but can potentially also reduce the safety requirements for this mapping since each medical center would have to take required precautionary measures by themself. As already covered, an irrecoverable mapping would not only impede the collection of further research data but also prevent patients from executing their data subject rights.

- Another idea would be to store the mapping in the medical center's client software, but in this case similar considerations like the ones given in Section 6.2.4 regarding cryptographic keys would impair the safety requirement. Since we developed the software component in the form of a web application, as justified in Section 6.5, there are no reliable places to safely and securely store the mapping.

- A final possibility would be to rely on a solution for distributed pseudonymization, such as the one covered in Chapter 3. The medical center would request the pseudonym for a patient, indirectly identified by their personal data, from the pseudonymization service and use it to store medical data with this pseudonym in the platform. Combining this with a solution for being able to disclose pseudonyms in a protected manner, such as the one presented in Chapter 4, this would also enable features such as the execution of data subject rights. But in this case a committee would have to disclose the patient identity for each data subject right request. Additionally, this solution would involve at least one other party which would increase the attack surface of the system. Furthermore, the mentioned convenience features would be not possible.

In our opinion, our concept based on cryptographically enforced client separation covered in Section 6.2.3 represents the best balance between the safety, security, and usability requirements, as it behaves like local storage of the pseudonym mapping through encryption, but also provides safety properties in control of the platform, while still enabling useful platform features.

6.2.2 Adversary Model

In this section we describe the adversary model of our platform, that is, the types of adversaries the platform protects collected data against. We assume secure point-to-point connections between hospitals and the central server. Eventhough in their absence the personal data would still be protected as it is encrypted directly at the hospital, the access to medical data during transport would tamper with further protection mechanisms like just allowing access through the privacy-preserving data interface (see Section 6.4). Furthermore, we assume the well-established ciphers which we use as building blocks in our platform to be practically secure when used with keys of adequate length.

With respect to the protection goals of information security *confidentiality*, *integrity*, and *availability*, the platform focusses especially on *confidentiality*. Protecting patients' personal and medical data is the first priority. A breach of confidentiality, for example through an unauthorized person linking medical data to a data subject, can have serious consequences for those affected (see Section 2.1). The integrity of medical data plays an important role as well for the medical study using this data, but cannot result in direct negative consequences for patients. The availability of the platform is not important. Medical centers can transfer collected data at a later point in time if the platform is not accessible.

With these considerations in mind, we look at potential adversaries the platform should protect against – within and outside of the platform user base. *Outside adversaries* with no relation to the platform must not able to access any data, let it be personal or medical, in unencrypted or even encrypted form. *Developers* of the platform must not be able to access any data in unencrypted form. The access to encrypted data records in the database during maintenance tasks might be necessary due to practical reasons, for example, the necessary debugging of bugs not reproducible on test systems. *Server admins*, operating the hardware the central component of the platform is deployed on, might get access to unencrypted medical data. This is not avoidable since the platform performs operations on this data, such as providing researchers with results to specific research queries in a privacy-preserving manner (see Section 6.4). Since the platform is expected to be deployed in hospital data centers, in which a high security level is present, the risks from this access are tolerable. Additionally, adversarial actions by developers or server admins might at least be detectable through adequate logging mechanisms. *Researchers* must only get access to medical data which was altered in a way to protect the privacy of patients. *Monitors* need to access medical data for a patient sample to review the correctness of this data. For this purpose, they also have to connect this data with patient identities to compare the data to health records in the medical center. However, this mapping should only be provided to the monitor in the (physical) medical center to reduce the probability of adversaries accessing it. The *survey admin* might be allowed to access collected medical data depending on the trust model employed for the medical study. If they are fully trusted, complete medical data access might be granted, for example, to perform data-based research or reviewing tasks on their own. Otherwise, they can simply take on an orchestrating role for the study without having access to

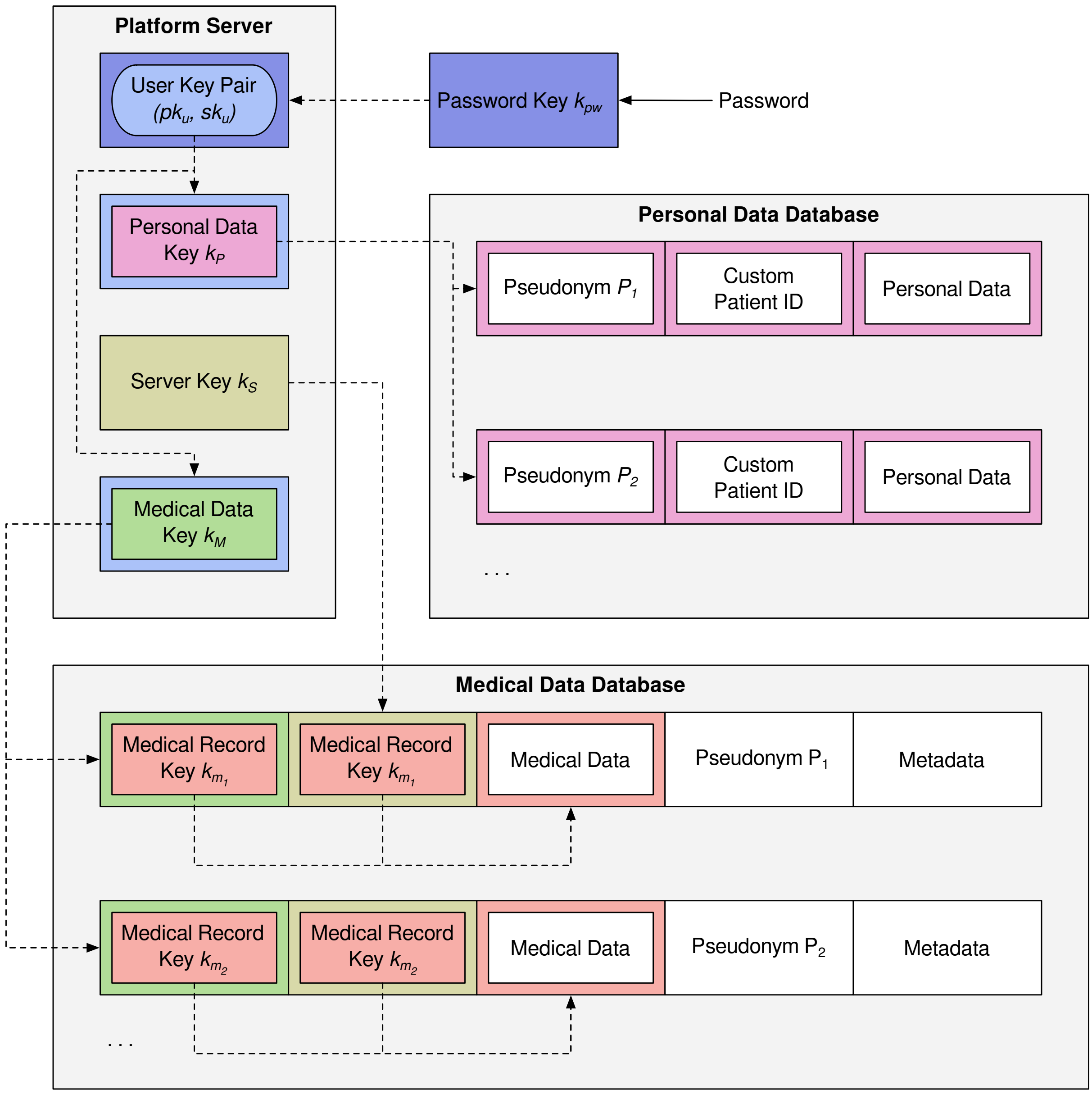


Figure 6.2: Overview of cryptographic keys and encrypted data in the platform.

medical data. Finally, medical center users (*medical center admin*, *doctor*, and *study nurse*) have access to medical and personal data of their own patients. In particular, they are the only parties to have access to unencrypted personal data of patients treated or examined at their medical center. They must not be able to access medical or personal data of other medical centers.

6.2.3 Cryptographically Enforced Client Separation and Pseudonymization

The platform stores the personal data for patients of a participating medical center in a way so that no other party, such as other medical center users, survey admins, or developers, can access this data. For this functionality we employ *cryptographically enforced client separation* and *pseudonymization*. Figure 6.2 shows the relevant cryptographic keys and encryption relations for one user of a specific medical center.

Each medical center user (study nurse, doctor, admin) chooses their own password pw. Then, a key derivation function (KDF) derives the password key $k_{pw} = \text{kdf}(pw \parallel \text{"key"})$ from this password. The concatenated label is used to separate this key from the authentication token discussed in Section 6.2.4. The password key k_{pw} is employed to encrypt the user key pair (pk_u, sk_u), which is generated in the medical center environment when a user logs into their account for the first time.

For each medical center there is a personal data key k_P for the encryption of personal data and a medical data key k_M for the encryption of medical data. These keys are generated in the medical center environment the first time the initial medical center admin logs into their account. All users in the medical center share these keys and they are encrypted with the help of the user public key pk_u for each respective user.

Equipped with these keys we can look into the details of encryption and storage of personal and medical data. The personal data of a patient together with a randomly created pseudonym P_i is encrypted with the personal data key k_P and stored in the personal data database. Since k_P is only accessible for medical center users (via decryption with their private key sk_u) the personal data of patients can only be accessed by users of this medical center.

A medical record key k_{m_i} is generated and used to encrypt the medical data of patients. This key itself is encrypted twice: once with the medical data key k_M and once with the so-called server key k_S. This server key is generated during the initialization of the platform. Encrypting the medical record key with distinct keys is essential for enabling the medical center to access their patients' medical data, while also allowing the platform to utilize this data for research purposes (as discussed in Section 6.4) and additional purposes such as benchmarking (see Section 6.1.4). This approach also forms the basis for restricting access to the medical data during maintenance tasks (refer to Section 6.2.7). A tuple consisting of the unencrypted patient pseudonym P_i, the encrypted medical data, both encryptions of the medical record key k_{m_i}, and additional metadata is stored in the medical data database.

Each medical data record belonging to a patient gets assigned the same pseudonym, making this a *person pseudonym* and the process deterministic pseudonymization (see Section 2.4.2). The resulting linkability is a requirement from the application scenario. Longitudinal medical studies require the connection of data collected at different points in time, for example, initial treatment data and data from subsequent follow-up examinations.

6.2.4 User Authentication

As already described, the platform uses the user password pw for deriving the password key k_{pw} (see Section 6.2.3). Additionally, the password is used to generate a token for user authentication as well. For this purpose, the password pw is given to a KDF in the medical center client software. The resulting value $l_{pw} = \text{kdf}(pw \parallel \text{"login"})$ is treated as authentication token and is stored in the platform database for the user account in once more hashed format (following password storage best practices). The concatenated label, again, is for separating the authentication token from generation of the password key k_{pw} discussed in Section 6.2.3. Due to this approach, the highly sensitive password is only known to the client software, which derives password key and authentication token, but is never transmitted to the central component.

On the other hand, this still opens up the potential for brute-force attacks on the authentication token to figure out the password, derive the password key k_{pw}, and consequently get access to the patient's personal data.

A more secure approach would be to use an independent encryption key instead of one derived from the password. There would be different options for this, but they come with practical downsides or impossibilities in our scenario:

- Two independent passwords could be used – one for authentication[2] and another one for deriving the password key. But this would require users to memorize two passwords and enter them during or after login. Since choosing and memorizing a single password is already a major challenge for users (see also Section 6.5), two passwords could potentially even weaken the security, for example, by encouraging users to write down both passwords on sticky notes pinned to their monitor.

- The same consideration applies to splitting the user provided password in halves, since this would require unusually long, harder to memorize passwords for sufficiently secure password halves.

- Another possibility would be to generate a random cryptographic key in the place of our password key and store this key in the central platform component with the user account data. But storing this key in unaltered form would render the whole concept useless against anyone with access to the database. Storing the key in encrypted form would just open up the question of which key to use for this encryption.

- Another option would be to generate and store the password key in the medical center client software only. This would be a favourable solution if there was a dedicated client software for the platform. However, as detailed in Section 6.5, the platform's client software is developed as a web application. Keys could be generated in the medical center client software, but would have to be stored in unreliable places like the browser cache, which are suspectible to data loss. Additionally, they somehow would have to be transferred to all devices of a user in the medical center without support of the platform.

- Finally, one could use suitable hardware tokens to generate and store the key pair (pk_u, sk_u) for each user and perform the decryption operation without exposing the private key to any platform software component. From a security standpoint, this would be a great solution. However, this would require every user of the platform to be equipped with a hardware token. This would not just be a financial problem, but also a practical one, since often computers in medical centers have strict policies against any additional external devices.

Therefore, the current way of using the password for both operations is the only viable solution. By using a memory-hard, state-of-the-art KDF like *Argon2* [Wet16], strong passwords, and two-factor authentication (see Section 6.2.5) the success probability of brute-force attacks is minimized.

[2] One could derive an authentication token as already described or just sent the plain password like it is state-of-the-art in regular web applications.

6.2.5 Strong Passwords and Second Factor Authentication

As Section 6.2.3 indicates, the user password plays a vital role in the security of patients' personal data since the relevant personal data key k_P used for encrypting personal data is encrypted with a user's secret key sk_u, which in turn is encrypted with the password key k_{pw} directly derived from the user password. Therefore this password serves as a security anchor for the most sensitive data in the platform. But a large amount of real-world incidents[3] shows that regular users are inherently bad at choosing secure (that is, high-entropy) passwords. To impede the vulnerability arising from weak passwords, the platform uses a password strength estimator and two-factor authentication (2FA). This twofold approach ensures two things. First, it prevents brute-force attacks by outside adversaries against the password through the regular authentication interface. Secondly, it impedes brute-force attacks against the authentication token or the encrypted user key pair by adversaries with access to the database.

For password strength estimation we employ the free software `zxcvbn` [Whe16]. `zxcvbn` is not based on simple heuristics like counting the occurrences of upper- and lowercase letters, digits, and symbols. Instead it uses various data sources (leaked password sets, common names, and common words), variations and combinations of words in these data sources, as well as keyboard patterns to estimate the commonness of a password. This leads to better strength estimation in comparison to simple heuristics.

We further use time-based one-time password (TOTP) [MRa+11] as a method for 2FA. TOTP is an HMAC-based one-time password algorithm depending on a secret K shared between *prover* (the user in our setting) and *verifier* (the platform server) and a time value T derived from Unix time and a time step parameter:

$$TOTP(K,T) = \text{Truncate}(\text{HMAC-SHA-1}(K,T))$$

On the first login the platform provides the user a randomly generated shared secret K and stores it in its user database. The user stores this secret on a device other than their device used for accessing the platform, such as their mobile phone. On subsequent logins the user computes the TOTP value on their second device, submits it together with their regular login information, and the platform verifies it by comparing the value with its own computation. Since TOTP values are based on the current time, they are short-lived and a loss of these values does less harm than the loss of long-lived secrets. Using another device as a second channel increases the security of the technique.

6.2.6 Key Safety

In addition to security aspects, safety aspects play an important role as well. If the personal data key k_P of a medical center is no longer accessible, all personal data cannot be decrypted (or, slightly less seriously, has to be re-entered and matched to the correct medical data records if the documentation still exists). If the server key k_S gets lost, no medical record key k_{m_i} and therefore no medical data is accessible for the platform anymore. In this case, all medical centers would have to decrypt their copy of k_{m_i} for

[3] For example, in 2023 the most common user passwords obtained from data breaches and stealer malware still included passwords like *password*, *123456*, and *qwerty* [Nor23].

their medical records and the platform would be required to encrypt all these keys with a new server key. In summary, loosing access to these keys has severe consequences for the availability of research data.

For this reason the platform provides users the opportunity to create physical backups of relevant keys in the form of printed quick-response codes (QR codes). This refers to the server key k_S for the survey admin as well as the personal data key k_P and the medical data key k_M for medical center admins. Additionally, each user in a medical center has this option for their secret key sk_u. By using a regular webcam, a user can conveniently restore respective keys using these QR codes without having to correctly type cryptographic keys[4].

6.2.7 Maintenance Mode

To prevent developers or server admins from accessing the medical data during maintenance operations like software updates or similar tasks the platform provides a maintenance mode. The survey admin can activate this mode before maintenance takes place. This deletes the server key k_S – one of the two keys used for encrypting the medical record key k_{m_i} (see Section 6.2.3) – from the platform. The survey admin is responsible for storing it safely during the time of maintenance. Afterwards they can re-introduce this key to the platform and regular operation can proceed.

At the time of writing the maintenance mode is just implemented as an organizational measure, that is, the survey admin is required to start this mode before developers enter the platform server, but it is not technically enforced. A more sophisticated solution could connect the activation of the maintenance mode with the server access technology, for example, by just starting the secure shell (SSH) server, which is used for developer access to the platform server, after the server key is deleted.

6.2.8 Reducing the Re-Identification Risk from Medical Data

While the platform prevents the access to patients' personal data for users not related to the patients' medical centers, it might be possible to re-identify the patients based on their medical data. While this risk cannot be prevented generally for arbitrary data (cf. Sections 2.6.1, 2.6.8 and 2.8), we introduce two measures to prevent common means of re-identification.

The first is tackling re-identification through information provided in free text fields. Medical center users can inadvertently enter data which supports the identification of individuals, such as names or gender information not being part of the medical data. This data can be used for re-identification purposes. To prevent these mistakes, users are initially trained to not enter directly identifying information in free text fields. Additionally, all text fields are marked with a prominent warning.

[4] For example, for common cryptographic key lengths of 256 bit this would require a user to correctly type 64 characters (when using hexadecimal encoding), which one would not directly consider usable software.

Another option for easy re-identification is to use time data and combine it with real-life observations, such as witnessed hospital visits by acquaintances. Therefore, all time-related data regarding patients, such as the treatment date or dates of follow-up examinations, is shifted by a random amount of days with respect to the original point in time. For this purpose each patient's personal data contains a shifting value t_S drawn uniformly at random from a span of 30 days. All time information in the medical data of a patient is shifted by this value t_S. Using the same value keeps all time spans between relevant medical events consistent, which is important as they can provide valuable insights for medical research, for example, when studying the occurrence of drug side effects. Because t_S is part of the personal data, it is encrypted in a way that only the patient's medical center can access the unencrypted value (see Section 6.2.3) and re-shift the time information. The platform handles this fully transparent to the medical center users. This measure is meant to impede adversaries with (legitimate or unauthorized) access to the medical data from connecting medical data records to patients via real-world observations.

6.3 Patient-Provided Medical Data

An additional feature of the platform is the possibility to let patients provide self-collected medical data. Examples for the usefulness of this feature include daily questionnaires about data attributes they can assess by themselves (often referred to as patient-reported outcome measures (PROMs)) or data collected by wearables. As a proof of concept, we have developed a mobile app for collecting quality of life data. This type of data can include daily data points about, for example, pain, physical abilities, or general life satisfaction.

To transfer the collected data to the database, an application programming interface (API) endpoint accessible from the internet is provided. Since the personal data in the database is encrypted and not accessible for the platform, we have to rely on the patient pseudonym P_i (see Section 6.2.3) to assign the patient-provided data to the correct patient data record. For this purpose the patient receives a copy of this pseudonym as a QR code and scans this code with the developed mobile app. The API request contains the pseudonym and new medical data can be assigned to the correct data record.

To prevent misuse of the public endpoint, the request must contain authentication information. Even though the pseudonyms have a length of 256 bit and are therefore large enough to act as authentication token by themselves in principle, a lost copy of the QR code would allow anyone to act as the patient. For this reason, we use the patient's birth date d as a second authentication factor not included in the QR code. When the QR code is created in the medical center, the platform client chooses a random value $r \in \{0,1\}^n$ and computes the authentication token $t = H(P_i) \oplus H(d) \oplus r$ using a cryptographic hash function H, such as SHA-3, with an output length n. Since the medical center is able to decrypt the personal data, they have access to the contained birth date. The value r is included in the QR code data, but not stored anywhere else. The token t is then stored in the database accessible for the platform. Because of the random value r the platform is not able to achieve information about the patient's birth date even when performing a brute-force attack using all reasonable birth dates. The patient has to enter their birth date in the mobile app as well. Afterwards, the mobile app is able to compute the authentication token t' using the provided birth date and the information given by the QR code (pseudonym P_i and random value r) in the same manner. This token is then provided in API requests. The platform compares the received token value t' with their stored token value t. Equal values result in an authenticated request.

Lost or compromised QR codes can be handled by simply deleting the stored token t and providing the patient with a freshly generated QR code using another random value r. To prevent brute-force attacks against the birth date value by an adversary in possession of the QR code, the token is invalidated after a specifiable number of requests with an invalid token. Using the birth date as second factor instead of for example a fully random token represents a compromise between security and usability. The second token would have to be transferred to the patient by other means (such as postal delivery) increasing the required effort and potentially time for authentication. Additionally, this would increase the practical risk that both factors would be stored side-by-side by the patient.

Currently the API endpoint is only allowed to write data to the patient's data record. Even if an adversary gets access to the authentication information, the only adverse action is to provide incorrect medical data. Conspicuous information might attract attention of staff in the medical center and they are able to delete illegitimate data and invalidate corrupted authentication information.

A future functionality for the mobile app worthwhile for patients would be the possibility to exercise their data subject rights (see Section 6.1.5) in the app. Due to the increased severity of these actions (for example, the option to get a full copy of all collected data) more security measures should be considered in this case.

6.4 Privacy-Preserving Data Interface

The platform provides an interface to grant researchers access to the medical data in a privacy-preserving way. The goal is to provide first insights at a low barrier with small privacy risks, not to provide a fully functional data analysis platform. For this purpose, the interface employs syntactic privacy models like k-anonymity (see Section 2.6) for exporting transformed datasets, differentially private statistics with respect to certain aspects of medical data, as well as an interactive query-based approach dependent on DP (see Section 2.7). While further users could potentially benefit from such an interface, we explicitly provide it for accounts with the *researcher* role to be able to establish a strict privacy model. Researchers can use the interface to get access to medical data tailored to their research needs without the necessity of full medical data disclosure. We further designate the survey admin as responsible for providing scenario-specific properties, such as the classification of attributes for syntactic privacy models or the parameter values for DP.

Generally, the interface can be used in different ways:

- Survey admins provide global datasets transformed by syntactic privacy models.

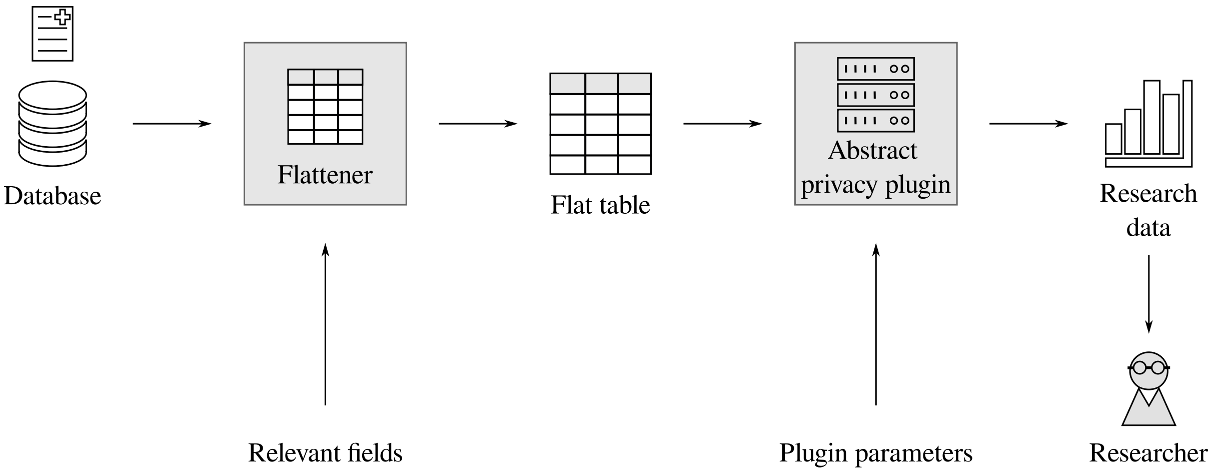


Figure 6.3: The privacy-preserving data interface architecture with an abstract plugin.

- Survey admins provide differentially private global statistics, for example, demographical data distributions of all patients in the platform.
- Researchers request specific datasets transformed by syntactic privacy models.
- Researchers ask the platform for specific computation results tailored to their research needs in a differentially private manner.

Further methods, such as using simple de-identification techniques (see Section 2.5), are possible but not covered in this section. While the other sections of this chapter describe mechanisms deployed into practice, the privacy-preserving data interface has not been utilized in the medical studies the platform has been used for. Therefore, the ideas are not practically evaluated by regular users of the platform at the time of writing. The following sections provide details on the general plugin-based interface architecture as well as implemented plugins based on syntactic privacy models and DP.

6.4.1 Interface Architecture

In the platform, the medical data is available in the form of a relational database: there exist multiple tables connected by primary keys and potentially multiple records and subrecords per patient. The first step in the process is to map this relational data into a flat table usable for further processing steps. For this purpose, the querying party, that is, researcher or survey admin depending on the scenario, chooses data fields relevant to the query. This table comprises exactly one row for each patient to avoid ambiguity regarding the influence of a single person to the data. If there are varying numbers of data records for patients[5], the flattened table contains as many columns for these records as are required for the patient with most data records. Afterwards this flattened table serves as basis for the privacy-preserving transformation covered in the next sections. These transformations are implemented in a plugin-ready architecture style so that extensions for other privacy mechanisms are easy to achieve. Each plugin

[5] Examples include varying numbers of required operations or situations in which data records are entered for each day spent in the hospital after a treatment.

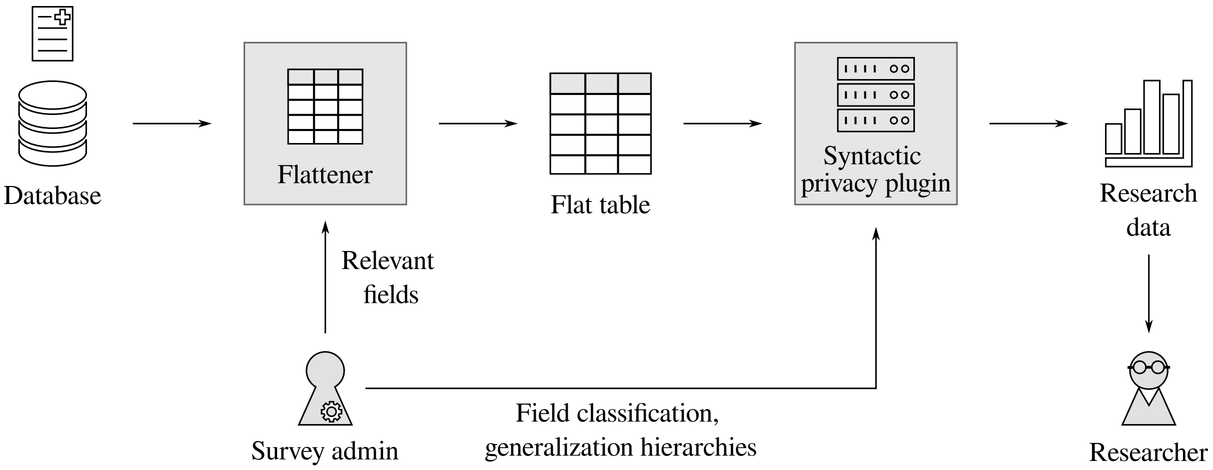


Figure 6.4: Privacy-preserving data interface using a syntactic privacy model and global (that is, survey admin controlled) data export.

takes the flattened table and a plugin-specific set of variables and produces some output depending on the mechanism. The abstract architecture is depicted in Figure 6.3.

6.4.2 Syntactic plugin

As stated in Section 2.6.9 syntactic privacy mechanisms suffer from several weaknesses. However they might serve the purpose of keeping honest people honest, as Narayanan and Shmatikov [NS19] mention for the case of de-identification techniques, if researchers are trusted to some extent. For this reason, we implement an instantiation of our plugin using k-anonymity (see Section 2.6.2) as syntactic privacy mechanism.

Different Modes

The syntactic privacy mechanism can be used in two ways. First, the survey admin can prepare *global* syntactically transformed tables accessible for all researchers. This approach is depicted in Figure 6.4. The survey admin selects the relevant fields for each table and classifies the fields according to their sensitivity into identifying, sensitive, QID, or nonsensitive attributes (see Section 2.6.1 for comments). Generally, we don't expect any attributes classified as identifying due to the separation of medical and personal data covered in Section 6.2.1. For a syntactic algorithm based on generalization (see Section 2.5.1) the survey admin provides adequate generalization hierarchies for QID fields. Afterwards the plugin computes the syntactic privacy algorithm result. Special care has to be taken when creating multiple tables containing data of the same patients or when tables are updated over time since these can lead to vulnerabilities (see Section 2.6.8).

In the second usage scenario, individual researchers can prepare their own *local* data exports according to their research needs as shown in Figure 6.5. This idea can have some advantages in comparison to the global approach. Researchers can tailor the resulting table to their needs by only choosing fields relevant to their research. This follows the idea of data minimization: Unnecessary data is not processed and released

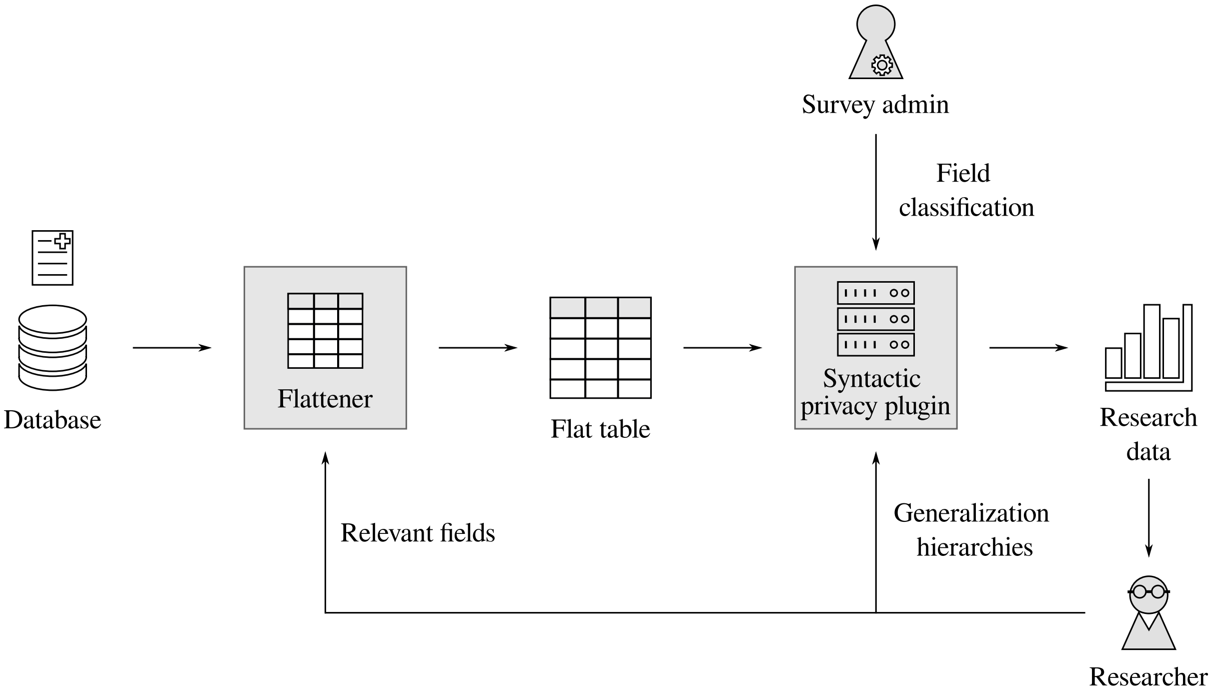


Figure 6.5: Privacy-preserving data interface using a syntactic privacy model and local (that is, researcher controlled) data export.

preventing the negative privacy impact this data might entail. Additionally, fields classified as QID can potentially require less distortion, such as generalization, to fulfill the syntactic privacy model if other QID attributes are not requested by the researcher. As an additional advantage, researchers can provide their own generalization hierarchies suiting their research needs better than global hierarchies. For example, some research might require ages in 5 year bins, while for others only the distinction between underage persons and adults might be relevant. In this scenario, the survey admin still has to classify the attributes just like in the global case.

But this scenario requires additional assumptions. We assume that researchers eligible to receive the transformed datasets do not have access to unaltered medical data, for example, by being member (study nurse, doctor, or medical center admin) of a participating medical center in addition to their role as a researcher. Furthermore, we assume that researchers do not share the transformed datasets with each other. Otherwise the mentioned vulnerabilities with respect to publishing multiple tables apply as well.

Plugin implementation

We have implemented one variant of a plugin targeting k-anonymity as syntactic privacy model. The plugin is based on an existing implementation[6] of the *Mondrian* algorithm covered in Section 2.6.5.

[6] The implementation of the Mondrian algorithm is based on `https://github.com/kaylode/k-anonymity` which updated the implementation of `https://github.com/qiyuangong/Basic_Mondrian` to be compatible with Python 3.

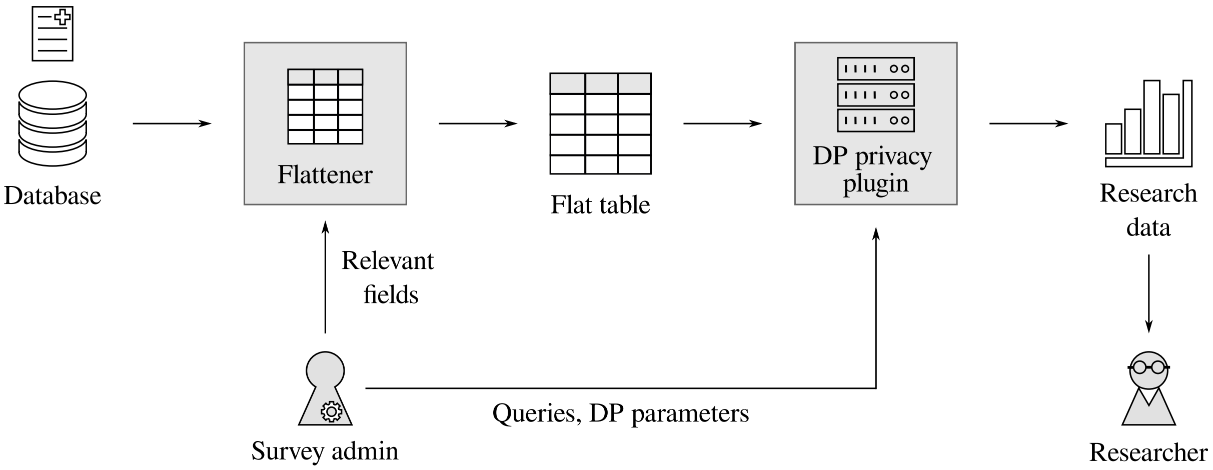


Figure 6.6: Privacy-preserving data interface using a differentially private privacy model and global (that is, survey admin controlled) data export.

Depending on the scenario, required parameters are prepared by the survey admin and researchers. In both scenarios, the survey admin classifies the attributes – in the global case when preparing the data export, in the local case when setting up the system. For the global case, they also provide generalization hierarchies for fields classified as QID. For the local case, they can prepare default generalization hierarchies during systems setup. These hierarchies can be used by researchers, but they have the opportunity to provide their own hierarchies when requesting a data export as well.

6.4.3 Differential Privacy Plugin

As detailed in Section 2.7.7, DP provides several useful properties as a privacy model. Especially the composability properties (see Section 2.7.2) allow the execution of multiple queries by multiple users while preserving privacy guarantees. This is a useful property in our scenario in which a set of researchers are allowed to query the platform interactively and according to their research needs (*local* approach). Additionally, the plugin can be used in a *global* fashion, in other words, the survey admin prepares the exported universal statistics for all researchers.

For the implementation of this plugin, a number of potentially usable DP frameworks was developed in recent years. Therefore, this section also contains a review of relevant DP frameworks before presenting the plugin implementation based on the chosen framework.

Different Modes

As already mentioned in the beginning of this section, DP can be applied in two different ways in our scenario.

First, DP can be used to provide global statistics about the patients and medical data in the database. This approach is depicted in Figure 6.6. These statistics are not dependent on specific research queries and can be accessed by all researchers. They can provide

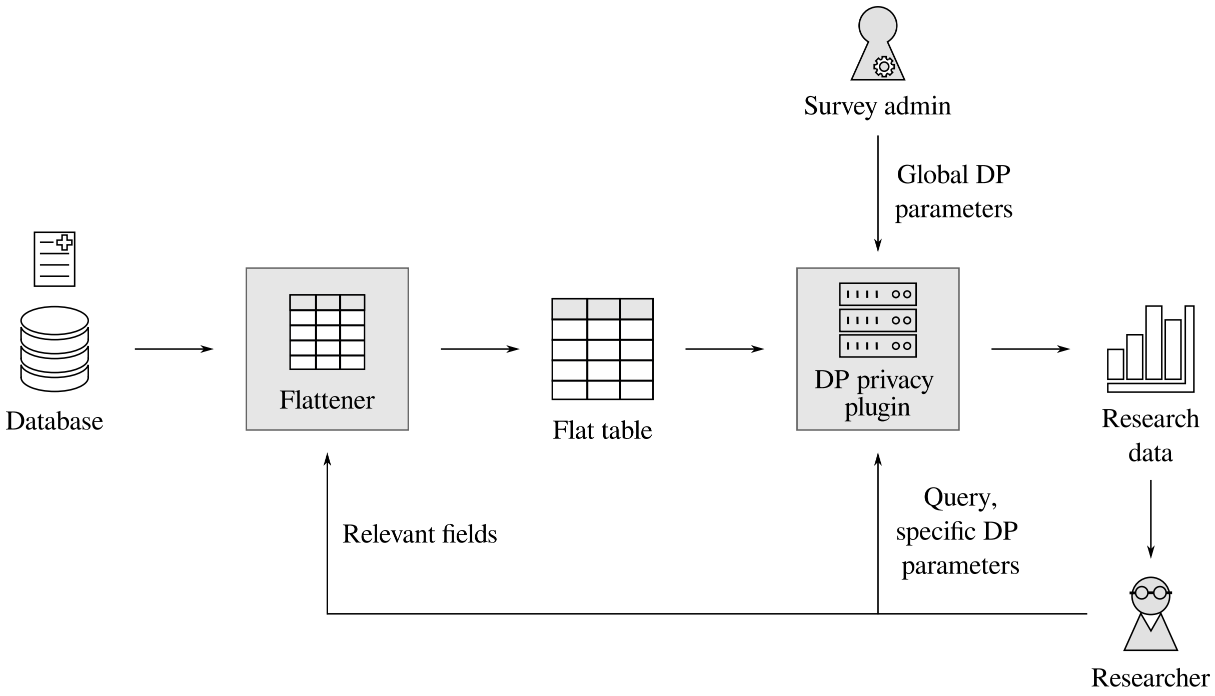


Figure 6.7: Privacy-preserving data interface using a differentially private privacy model and local (that is, researcher controlled) data export.

general insights about the medical data, for example, demographical distributions like the patient age distribution, and might prove useful as a starting point for further, more specific research. The survey admin chooses relevant data fields, relevant statistical queries, such as simple counts, histograms, or percentiles, and adequate DP parameters, for example, ε and δ in the case of (ε, δ)-DP. They split the global privacy budget between the queries according to utility requirements of respective queries. This action requires a good understanding of DP and the influence of parameter choices on query results. A helpful technique, currently not implemented in the platform, to support regular users in choosing suitable parameters might be to provide the accuracy implications in the form of confidence intervals [Gab+18]. If regularly updated statistics are desired, the budget splitting must take this requirement into account.

The second usage scenario, visualized in Figure 6.7, focuses on interactive queries by researchers. The platform provides a query interface and researchers choose relevant data fields, query mechanisms and DP parameters according to their research needs. In this scenario, there potentially are multiple researchers and each of them might have multiple queries asked in an iterative fashion, in other words, the next query depends on the answer of the former query (also referred to as *online* setting [Puj+21]). This can pose a major challenge for allocating the privacy budget among researchers and queries. Questions to consider here include amongst others [ico22]: How many queries are expected in total and per researcher? Which budget do these queries need to achieve necessary utility requirements? Are researchers expected to (maliciously or unintentionally) share results of their queries?

In our solution, we mostly follow the ideas of Gaboardi et al. [Gab+18] with respect to budget splitting. The survey admin, just like in the case of global statistics, chooses an adequate global DP privacy budget, for example, ε for ε-DP. Researchers are classified according to two trust models. *Semi-trusted* researchers are assumed to be trustworthy in not sharing their query results with other parties. This might be based on personal trust, contractual agreements, or other trust-building measures. This model allows the platform to assign them the full privacy budget (ε in the example). The other model assumes *untrusted* researchers, for which the non-sharing cannot be guaranteed. In this model each researcher only gets a share of the global privacy budget for their queries, for example, each of the n researchers receives a privacy budget $\varepsilon_i = \frac{\varepsilon}{n}$. In this setting it is also possible to publish all query results publicly for all researchers. The two models are not mutually exclusive. It is possible to reserve a part of the privacy budget for semi-trusted researchers and to split the remaining budget among untrusted researchers.

There are results for splitting the privacy budget among multiple parties to achieve better utility in comparison to the simple split considered above [Puj+21], but to the best of our knowledge they only consider the offline setting, in other words, non-iterative queries.

Existing Differential Privacy Frameworks

In recent years, a variety of DP programming frameworks has been developed. These frameworks aim to simplify developers' access to DP methods while hiding the complexities of DP mechanisms through user-friendly interfaces. This section offers a comprehensive overview of these frameworks. To collect relevant frameworks we have used a web search with *startpage* as well as several more and less formal surveys [Far23; Woo+23; Dif23; Gar+23] and reviewed the union of resulting frameworks.

In this work our focus is on general-purpose frameworks usable for arbitrary queries. Several frameworks are tailored towards specific goals and are therefore considered out-of-scope in this section. This involves a class of frameworks specifically developed for ML applications. These frameworks include *RAPPOR*[7] [EPK14], *PyTorch Opacus*[8], *TensorFlow Privacy*[9] [Aba+16], and *PySyft*[10]. Another class of frameworks focus more on specific aspects, such as the verifiability of DP properties. These frameworks include *εktelo*[11] [Zha+18], *Chorus*[12] [Joh+20], *Duet*[13] [Nea+19], *Fuzzi*[14] [Zha+19], and *LightDP*[15] [ZK17]. Other tools provide visual support for the application of DP and the selection of required parameters. These tools were already covered in Section 2.7.5.

[7] The code repository is available at `https://github.com/google/rappor` (visited on 14.11.2023).

[8] The code repository is available at `https://github.com/pytorch/opacus`, the website at `https://opacus.ai/` (visited on 14.11.2023).

[9] The code repository is available at `https://github.com/tensorflow/privacy`, the website at `https://www.tensorflow.org/responsible_ai/privacy/guide` (visited on 14.11.2023).

[10] The code repository is available at `https://github.com/OpenMined/PySyft`, the website at `https://openmined.github.io/PySyft/` (visited on 14.11.2023).

[11] The code repository is available at `https://github.com/ektelo/ektelo` (visited on 14.11.2023).

[12] The code repositories are available at `https://github.com/uber-archive/sql-differential-privacy` and `https://github.com/uvm-plaid/chorus` (visited on 14.11.2023).

[13] The code repository is available at `https://github.com/uvm-plaid/duet` (visited on 14.11.2023).

[14] The code repository is available at `https://github.com/hengchu/fuzzi-impl` (visited on 14.11.2023).

[15] The code repository is available at `https://github.com/yxwangcs/lightdp` (visited on 14.11.2023).

In contrast, several other frameworks have no special focus but target general applicability: *PipelineDP*, *Tumult Analytics*, *Google DP*, *PyDP*, *OpenDP*, *IBM Diffprivlib*, and *PinQ*. We describe these frameworks shortly in the following. Table 6.1 summarizes the covered frameworks, respective online resources, and relevant properties.

PipelineDP is developed in collaboration between *Google* and *OpenMined*. It is specifically targeted at large-scale data analytics engines like *Apache Spark*[16]. For this purpose PipelineDP provides interfaces encapsulating DP functionality specific for different data analytic frameworks. At the time of writing PipelineDP is still considered experimental and not recommended for production use[17].

Tumult Analytics [Ber+22] by *Tumult Labs*, just like PipelineDP, targets Apache Spark and can be used to analyze large datasets. It supports several aggregation functions and DP variants. Tumult Analytics is ready to be used in production and is already employed by the *Census Bureau* in the USA[18].

Differential Privacy [Wil+19] by Google is a collection of libraries related to DP. In addition to the DP building block library, which is relevant for our scenario, other libraries – amongst others one for large-scale data analytics on *Apache Beam*[19] and another one for differentially private database queries – are also present.

PyDP is a wrapper library by *OpenMined* to use the Google DP building block library with the Python programming language. PyDP offers a subset of all the functionality of Google's libraries.

The *OpenDP Library* [Tea20; GHV20] is part of the *OpenDP Project* originally founded at *Harvard University*. At the time of writing, the library still undergoes large development efforts and interfaces are likely to change[20].

Diffprivlib [Hol+19] developed by *IBM*. It provides several low-level mechanisms, functions for higher-level queries, as well as prepared differentially private ML models. At the time of writing, the library is still not present in a major version 1 and therefore likely to change.

Privacy Integrated Queries (PinQ) [McS09] developed by Frank McSherry at *Microsoft* is a framework providing basic DP functionality with a LINQ[21]-like interface. PinQ is based on an early research effort to develop usable DP frameworks and has not been updated for several years.

Plugin implementation

Based on the framework review we have decided to use *Tumult Analytics* for our DP plugin due to several reasons. In comparison to most other frameworks, this framework already has a stable version and is used in production. Since we have implemented

[16] `https://spark.apache.org/` (visited on 21.11.2023)
[17] `https://pipelinedp.io/overview/` (visited on 21.11.2023)
[18] `https://www.tmlt.io/` (visited on 21.11.2023)
[19] `https://beam.apache.org/` (visited on 21.11.2023)
[20] `https://docs.opendp.org/en/stable/user/limitations.html` (visited on 21.11.2023)
[21] *LINQ (Language Integrated Query)* is part of Microsoft's .NET framework and serves the purpose of data querying.

Table 6.1: Overview of existing general-purpose DP frameworks. URLs and repositories were last visited on November 14, 2023.

Framework	Web	Github	Last Commit	Languages	DP Variants	Queries
PipelineDP	https://pipelinedp.io/	https://github.com/OpenMined/PipelineDP/commits	2023-11-03	Python	ε-DP (ε, δ)-DP	Count, Mean, Privacy ID Count, Sum, Variance, Vector Sum, Percentile, Truncated Geometric, Laplace Thresholding, Gaussian Thresholding
Tumult Analytics [Ber+22]	https://www.tmlt.dev/	https://gitlab.com/tumult-labs/analytics	2023-10-30	Python	ε-DP (ε, δ)-DP zero concentrated DP	Count, Count Distinct, Quantile, Min, Max, Median, Sum, Average, Variance, Standard Deviation
Google DP [Wil+19]		https://github.com/google/differential-privacy	2023-10-26	C++, Go, Java	ε-DP (ε, δ)-DP	Count, Sum, Mean, Variance, Standard Deviation, Quantiles, Automatic Bounds Approximation, Truncated Geomatic Thresholding, Laplace Thresholding, Gaussian Thresholding, Pre-Thresholding
PyDP	https://pydp.readthedocs.io/en/latest/index.html	https://github.com/OpenMined/PyDP	2023-10-18	Python	ε-DP	Mean, Sum, Standard Deviation, Variance, Max, Min, Median, Count, Percentile
OpenDP [Tea20; GHV20]	https://opendp.org/	https://github.com/opendp/opendp/	2023-10-17	Rust, Python	ε-DP (ε, δ)-DP zero concentrated DP	Count, Sum, Mean, Quantile, Variance
IBM Diffprivlib [Hol+19]	https://diffprivlib.readthedocs.io/	https://github.com/IBM/differential-privacy-library	2023-08-31	Python	ε-DP (ε, δ)-DP	Count Non Zero, Mean, Standard Deviation, Sum, Variance, Quantile, Percentile, Median
PinQ [McS09]	https://www.microsoft.com/en-us/download/details.aspx?id=52363		2009-08-18	LINQ (C#)	ε-DP	Count, Sum, Average, Median

```python
def histogram_average_query(df: pd.DataFrame,
                            field_name: str,
                            group_by_keyset: KeySet,
                            lower_bound: int,
                            upper_bound: int,
                            epsilon: float) -> pd.DataFrame:
    session = Session.from_dataframe(
        privacy_budget=PureDPBudget(epsilon),
        source_id="patients",
        dataframe=df,  # dataframe contains flattened table of medical data
        protected_change=AddOneRow(),
    )
    query = (
        QueryBuilder("patients")
        .groupby(group_by_keyset)
        .average(field_name, low=lower_bound, high=upper_bound)
    )
    result = session.evaluate(
        query,
        privacy_budget=PureDPBudget(epsilon),  # consume full privacy budget
    )
    return result
```

Listing 6.1: An example function for the differentially private computation of averages for patients grouped by some criterion.

our platform based on python (see Section 6.5), using a framework directly supporting python allowed for easy integration. Finally, the framework provides several query types in an easy-to-use fashion.

We provide a fixed set of functions usable for the survey admin or researchers (depending on the scenario). Listing 6.1 shows an example plugin function for computing a differentially private histogram average query, that is, the average value for a specific field for patients grouped by another field value. This function can be used for queries like computing the average patient age for different cancer types in a cancer registry. In the displayed function we do not rely on the privacy budget accounting provided by Tumult Analytics, since we keep track of the budget accounting outside of the individual query functions. Some values have to be provided by the user performing a query. This includes the keys according to which the patient set is partitioned as well as upper and lower bounds for the required value clamping (see Section 2.7.3). In some cases the keys are provided automatically by the platform (for single-choice fields with a fixed value set), in other cases the user has to provide them (for example, adequate age bins). Similar functions are implemented for further query types, like count, sum, variance, or quantiles. Tumult Analytics provides further functionality like the option to filter data based on arbitrary criteria, which is not used in the platform at the time of writing but might be included in the future.

6.5 Platform Implementation and Lessons Learned

We have developed our platform in the form of a client (software component in the medical centers) and server (central component) architecture.

The server is implemented in python using the web framework *django*[22] and *django rest-framework*[23] for a REST-based API. This interface provides functionality for, amongst others, user authentication and management, storing and reading encrypted medical and personal patient data, as well as monitoring and benchmarking purposes. Connections to the interface are secured via Transport Layer Security (TLS) [Res18] allowing state-of-the-art CipherSpecs only.

For the client we implemented a browser-based solution (web application) in *elm*[24] – a functional programming language which is transpiled into JavaScript before execution. The web application JavaScript code is made available by the server API. After receiving this code, all medical center specific application logic, including security relevant operations like the encryption or decryption of medical or personal data, are performed by the medical center staff's web browser. The server has no access to locally processed data.

The decision to develop the client as a web application and not as a local application directly running on a machine in the medical center was a result of the application scenario. Computer systems in the medical sector are strictly administered and the installation or updating of software is often prevented or accompanied by a complex verification process. Since there were multiple medical centers participating in the medical studies, the client software would have to be installed and updated in each of these centers individually (if possible at all). This would have the potential to delay the inclusion of new centers as well as the deployment of functional and also potentially critical security patches significantly. The downside of this approach is that the platform code is always received from the central server each time a client uses the platform. Should an adversary gain access to the server, whether through being an authorized server admin or through an attack, they could potentially inject malicious client-side JavaScript code. This code could execute arbitrary actions, such as transmitting cryptographic keys or plaintexts of personal data to the adversary. Local applications are more secure in this regard. After they have been installed locally, an adversary, to perform a similar attack, has to target the local machine, which generally offers less attack surface than a semi-public server.

The cryptographic functionality used in the client and server is provided by *libSodium*[25]. This library uses state-of-the-art cryptographic algorithms and key lengths [Ber09]. It employs symmetric authenticated encryption based on the stream cipher *XSalsa20* and *Poly1305* for the MAC creation. For asymmetric encryption, the library utilizes a construction using ECC-based Diffie-Hellmann key agreement over *Curve25519*.

[22] `https://www.djangoproject.com/` (visited on 20.02.2024)

[23] `https://www.django-rest-framework.org/` (visited on 20.02.2024)

[24] `https://elm-lang.org/` (visited on 20.02.2024)

[25] *libSodium* (`https://doc.libsodium.org/`) is a fork of *NaCl* (`https://nacl.cr.yp.to/`) focusing on usability for software developers. Precisely, we use *PyNaCl* (`https://github.com/pyca/pynacl`) in server-side code and *libsodium.js* (`https://github.com/jedisct1/libsodium.js`) in client-side code. (visited on 20.02.2024)

The platform has been deployed and used in two real-world medical studies. In these studies the server component was deployed in a particularly secured area of a hospital data center. To reduce the attack surface, access to the API was protected based on internet protocol (IP) address filtering. Clients, that is, other medical centers collecting patient data for the study, initially had to provide IP address ranges for their institution. However, the extensive security and privacy measures employed in the platform and its deployment posed minor and major challenges for us. In the following we provide some anecdotal insights into these challenges.

As presented in Section 6.2.6, we provided a way to export backups of important cryptographic keys in the form of QR codes. The platform user interface notifies users explicitly about this function and the importance of being able to restore these keys. Despite these measures, we had to intervene during several occasions to help medical centers when their medical center admins had forgotten their passwords, had not created backups of their cryptographic keys, and therefore lost access to these keys. Our resolution involved temporarily elevating a regular user to a medical center admin to create a new admin account for the actual admin and revoke the rights afterwards. This solved the issue, given that at least one user account within the medical center had access to the necessary personal and medical data keys k_P and k_M (see Section 6.2.3). However, in one case where both existing users managed to forget their passwords, we were forced to create a new medical center and empty dummy data records for all patients' personal data. Then we could migrate all medical data records by decrypting the medical record keys k_{m_i} with the server key k_s and encrypting them again with the new medical data key k_M (in an automated fashion without us being able to learn the server key). Finally, the medical center user had to enter the personal data again after re-identifying the patients by their medical data.

The employed cryptographically enforced client separation (see Section 6.2.3) allows only medical center users to access the personal data of their patients. While this mechanism is crucial for the privacy guarantees of our platform, it poses challenges when necessary updates to the data model during iterative system development require modifications to the stored data. To address these challenges, we had to develop complex client-side migration processes to align the stored data with the updated data model. These migrations had to be executed in the user's browser after they logged in, since only then the personal data had been decrypted and was available for editing. A lot of thought was devoted to a robust design of the migration logic, so that even inadvertent user actions, such as closing the browser during the execution of a migration, do not result in data integrity issues or leave the data in an inconsistent state.

Another challenge arose from the constraint that we as developers were not allowed to examine the medical data, which was implemented in the maintenance mode (see Section 6.2.7). At one point in time we received a bug report relating to a specific medical data record. Even with extensive support of respective user, we were not able to reconstruct the problem in our test environment. Therefore, we had to perform cumbersome live debugging in collaboration with the user involving multiple tedious deployments of new platform versions before catching the bug.

A final challenge consisted in the high security requirements of the server environment our platform was deployed to in the hospital data center. To connect to the platform server we had to tunnel our network traffic through a reverse proxy, access to which was only enabled after contacting the data center employees via telephone. Unfortunately, the availability of employees has not always met our demands. This not only postponed several regular updates, but it also caused one major downtime of the platform when a bug fix could not be applied in time.

In conclusion, it is necessary to thoroughly evaluate the real-world implications of security and privacy measures, especially in unforeseen scenarios that may deviate from regular operations.

6.6 Related platforms

There is a large number of medical registries which target specific diseases[26] and a large share of these registries use self-developed software tools. Some companies offer commercial registry software, including BQS[27], RAYLYTIC[28], and IT-Choice Software[29]. Unfortunately, none of these companies provide technical details about security and privacy features of their solutions. But there are also some open-source platforms which, similar to our platform, are developed in a generic way to be used in a variety of scenarios.

The Collaborative Health Outcomes Information Registry (CHOIR) [Med24] developed by the *Stanford Pain Management Center* in partnership with the National Institutes of Health (NIH) is a platform which collects patient data to support clinical care as well as research. It additionally provides capabilities for further functionality like patient scheduling. Originally, it was developed in the field of chronic pain treatment. It is said to be open-source, but a code repository has not been made public at the time of writing. Also, there are only few details about the architecture of CHOIR given, so that we are not able to compare details to our platform.

The software Polymorphic encryption and pseudonymisation for personalised healthcare (PEP) [Ver+16; Uni24] developed[30] by the *Radboud University* which puts the data subject at control of their own data. Their data is encrypted in a way that they can later decide about which party is allowed to access and decrypt the data in an end-to-end fashion. PEP separates the responsibilities between three components (an encrypted data storage server and two complementary cryptographic key storage server) so that a single successfully attacked server still prevents unauthorized data access. Additionally, PEP provides a pseudonymization infrastructure to assign different pseudonyms to the same data subject for different parties. In comparison to our platform, PEP allows the patient more control about their data at the cost of requiring patients to approve each and every data export for researchers and potential further functionality like monitoring (see

[26] For example, the *registry database of medical registries in Germany* (german *Registerdatenbank der medizinischen Register in Deutschland*) [BQS24] contains 415 entries as of November 22th, 2023. About one third of the present registries use a self-developed platform solution and another third uses “standardized products” – details about these products are unfortunately not provided.

[27] https://www.bqs.de/bqs-register/02-bqs-register.php (visited on 16.02.2024)

[28] https://www.raylytic.com/loesungen/medizinische-register/ (visited on 16.02.2024)

[29] https://www.it-choice.de/produkte/starkit/ (visited on 16.02.2024)

[30] The PEP repository containing the documentation and deployable docker containers can be found at https://gitlab.pep.cs.ru.nl/pep-public (visited on 08.02.2024). PEP is announced as open-source in he future, but the source code is not published at the time of writing.

Section 6.1.3). This might reduce the amount of available research data, for example, in scenarios where patients have a high risk of deceasing due to their disease. Additionally, it is by no means obvious if protection measures like the DP-based data export described in Section 6.4 are even possible in PEP or if data users always would have access to the original data potentially putting patient's privacy at more danger.

The *Open Source Registry System for Rare Diseases* (german Open-Source-Registersystem für Seltene Erkrankungen (OSSE)) developed[31] by the *Goethe University Frankfurt* [Fra24] is an open-source registry software with a focus on rare diseases. Similar to our platform, OSSE allows to create scenario-dependent forms with various field types. In addition, the software includes a metadata repository allowing for interoperability between registries. To protect patient identities OSSE uses pseudonymization. For this purpose it relies on two distinct components, one to host the registry itself and one to host the pseudonymization service *Mainzelliste* (see Section 3.6). In comparison to our platform, in OSSE a component outside of the medical center – the pseudonymization service – gets access to unaltered personal patient data. A successful attack against this component (or a malicious insider) can put all patients at danger of connecting their medical data and their identity.

6.7 Conclusion

In this chapter we have provided a platform concept for privacy-preserving medical registries supporting longitudinal studies. The platform offers functionality for, amongst others, data quality validation through monitoring, performance comparisons between different medical providers, the execution of data subject rights, and the option to include patient-provided data. Personal patient data is protected by cryptographically enforced client separation so that this data is only ever accessible by the originating medical center, while medical data is stored in a pseudonymized form allowing to link patient's data records over time. The platform architecture is designed in a way so that even users with more access rights, like developers or the survey admin, cannot access the personal data. Further measures with respect to user authentication, password choices, 2FA, key safety, and re-identification risks support the privacy and security guarantees of the platform. To facilitate privacy-preserving access to collected medical data for researchers, a data interface was developed that ensures protected access using various privacy models, implemented in an extensible plugin-based manner.

One limitation of our platform was covered in Section 6.5. Since the client application was implemented as a web application being executed by a medical center's web browser, the application JavaScript code is always requested from the central server. An adversary with access to the server can always inject malicious code to leak cryptographic keys or plaintext personal data. A client application executed locally could prevent this weakness but would encounter practical difficulties (also mentioned in Section 6.5), especially in the medical domain.

One area for future research involves the enhancement of the data interface through the integration of additional privacy models. Moreover, the capabilities of the DP

[31] The code repository can be found at `https://bitbucket.org/medicalinformatics/mig.samply.edc.osse/src/master/` (visited on 01.02.2024).

plugin could be extended to provide more comprehensive functionalities. Specifically for researchers, an advanced query interface with extensive features could broaden the scope for investigating a wide range of research questions. Additionally, it would be essential to address issues related to the interface's usability for non-experts, for example supporting adequate privacy budget allocation.

7 Conclusion

The use of data-driven statistical research methods in the healthcare domain becomes increasingly prevalent and insights gained from personal health data allow for large advancements in the field. On the other hand, privacy concerns arise from the presence of this highly sensitive data. Data privacy techniques such as pseudonymization and anonymization are crucial for balancing the conflicting interests of data utility and privacy. Another dimension added to this problem is the distributed nature of potential research data, with various parties, such as hospitals, general practitioners, and insurance companies, being responsible for its collection and storage.

In this thesis we investigated the problem of implementing pseudonymization and anonymization methods in distributed environments in the healthcare domain. We started with a comprehensive overview of the field of privacy-preserving data publishing covering fundamental principles as well as different classes of techniques and also provided divergent positions of practitioners, legal scholars, and computer scientists. The remainder of the thesis was concerned with specific usage scenarios for pseudonymization or anonymization related methods in distributed environments. Based on the constraints of the scenarios, we designed and implemented several solutions to highlight the challenges and opportunities of distributed data privacy methods in the healthcare domain.

In this concluding chapter, we provide a summary of our contributions in Section 7.1, look at potential areas of future research in Section 7.2, and give a short outlook in Section 7.3.

7.1 Summary of Contributions

Five research contributions originated from this thesis. In the following we present the main results. We forgo reiterating limitations and future research possibilities, as they are discussed in the conclusion sections of respective chapters (Sections 3.7, 4.8, 5.5 and 6.7).

C.1: Technical and Legal Literature Review

Various risks for individuals arise from the exposure of personal data, ranging from stigmatization to direct physical harm. We have developed a taxonomy of disclosure risks emerging for individuals, whether included in a published dataset or not. This taxonomy unifies differing perspectives and terminologies found in the existing literature. Diverging terminologies are also evident for central terms such as *personal data*, *de-identification*, *pseudonymization*, and *anonymization*. We analyzed these terms from the perspectives of computer science as well as European and United States law and find significant regional and discipline-specific variations.

Furthermore, we investigated the central classes of privacy-preserving data publishing methods, namely pseudonymization, de-identification techniques, syntactic privacy models, and semantic privacy models. We presented detailed information on each method, encompassing fundamentals, basic techniques, advantages, and disadvantages. It becomes apparent that there is no one-size-fits-all solution in data privacy. Pseudonymization enables the linking of data records and re-identification, yet it does not prevent re-identification risks from data attributes that are not pseudonymized. De-identification techniques may offer some level of protection, but it is hard to evaluate their privacy impact. While syntactic privacy models offer quantifiable guarantees, the underlying principles such as the classification of attributes or assumptions about potential background knowledge cannot necessarily be maintained in practice. Semantic privacy models, particularly DP, achieve quantifiable and composable privacy guarantees without relying on assumptions about potential adversaries. However, these models are not suitable for analyzing small datasets and using them in practice entails several challenges, such as adequately choosing and distributing privacy budgets.

The varying perspectives on even fundamental terms and the absence of a universal solution for data privacy have raised emotional discussions on the correct way to handle data privacy among computer scientists, legal scholars, and practitioners. We summarized the debate, linked it to existing methods, and discussed potential policy changes to enhance the current legal status regarding data privacy.

C.2: Distributed Pseudonym Generation

We presented a solution for achieving pseudonymization in a distributed setting where data records about individuals are collected from multiple data sources. Our solution ensures that an individual is assigned the same pseudonym, regardless of the data source, a property we call *globally consistent pseudonyms*. Rather than relying on a trusted third party to maintain a global pseudonym mapping table in plaintext, our goal was to achieve this functionality without revealing the identity-pseudonym relationship to any party other than the data source. To achieve this, we employ a multi-reader multi-writer SE scheme to hide the identity data from a third party (the pseudonymization service), while still allowing searches for existing pseudonyms related to the identity data. Our work improves on related work by Zimmer et al. [Zim+20] in that it allows the management (enrollment and revocation) of data sources. Additionally, our work offers fuzzy search capabilities to link identities to pseudonyms in case of typos or variations in the spelling of identity data. Similar to the work of Zimmer et al., our solution enables the restriction of data record linkability with respect to time or budget restrictions to mitigate privacy implications to individuals when necessary. In our evaluation, we compared our scheme to several solutions proposed in related work regarding privacy-related properties as well as practical considerations. Our analysis revealed that no single solution fulfilled all the properties we defined. Therefore, users are required to select a solution based on the specific requirements dictated by their unique scenario. A performance evaluation demonstrated the practicality of the core scheme, with searches for 50,000 pseudonyms completed in under 0.3 seconds. However, the fuzzy search extension may be too resource-intensive in scenarios with even moderate pseudonym quantities, taking over 2 seconds for a database containing 1,000 entries.

C.3: Distributed Pseudonym Disclosure

In addition to the distribution of pseudonym generation we provide an approach for the distribution of pseudonym disclosure, that is, the re-identification of the individual related to the pseudonym. Since this re-identification can reveal highly sensitive information about an individual, safeguarding the disclosure process against unauthorized access is of high importance. To achieve this we enforce the so-called *multi-eye principle*, which requires the sensitive process to be approved by multiple parties. This principle is accomplished by utilizing threshold decryption, a class of cryptographic schemes which distribute the decryption process of a public-key scheme. By encrypting an individual's identifying attributes and storing the resulting ciphertext alongside the individual's assigned pseudonym, identity disclosure can only take place with the collaboration of a minimum of parties possessing shares of the respective private key. However, the management of keys in such schemes poses unique challenges compared to traditional public-key schemes. To address these challenges, we proposed a novel scheme for managing authorized parties in threshold decryption schemes employing PRE. This new scheme improves on related work in that its security guarantees do not rely on single parties deleting old key material or ciphertexts. We successfully implemented the scheme and validated its practical efficiency, achieving running times within the range of single-digit milliseconds for recurring operations.

C.4: Distributed Syntactic Anonymization

Shifting the focus from pseudonymization to anonymization, we investigated a protocol for distributed syntactic anonymization proposed by Mohammed et al. [Moh+10]. We identified vulnerabilities in the protocol stemming from the utilization of a basic *secure-sum protocol*. First, the protocol is vulnerable to colluding parties surrounding a victim within the communication ring structure it operates in. Second, a severe flaw allows the orchestrating party to deduce information about attribute distributions in the data – deductions which are not possible from the final protocol result, the table providing syntactic privacy guarantees. This leakage arises from the protocol's collection of sizes of potential equivalence classes, which would come into existence if further specializations of the data were carried out, under the condition that these equivalence classes meet the syntactic privacy requirements. Essentially, the protocol has to look one step into the future. In response, we provide an updated subprotocol based on SMPC which prevents the information leakage to mitigate the vulnerabilities. The basic idea of the subprotocol is to only compute if a potential equivalence class size exceeds a threshold, without disclosing the exact size. This protocol might be of independent interest for other researchers tackling related problems. We implemented the updated protocol and measured its performance. Due to the inherent computational and communication costs associated with SMPC, the updated protocol comes with a heavy performance penalty, ranging from 10 to 1000 times depending on the dataset and the number of parties. Despite the significant performance impact associated with the countermeasure, this should not generally prevent its implementation, as the protocol is typically intended for a one-time execution.

C.5: Privacy-Preserving Medical Registry

While the initial four contributions are more theoretical in nature, the final contribution deals with practical considerations around the practical implementation of privacy-preserving measures. Our objective was to provide a concept for a medical registry – a system to prospectively collect data on patients meeting specific criteria over a longer period. This registry should support longitudinal studies, which involve collecting data on individual patients at multiple points in time. Apart from this core feature, the registry should offer further functionalities such as monitoring the validity of collected data and comparing the performance of healthcare providers. We proposed an architecture incorporating various privacy and security measures to safeguard personal and medical patient data. These measures include pseudonymization, cryptographically enforced client separation, and additional techniques focusing on the security and safety of cryptographic keys. Additionally, we offered an extensible data interface that grants researchers access to medical data with syntactic or semantic privacy protections in place. The platform has been successfully implemented and utilized in two real-world medical studies involving over 5000 patients. We provide some insights gained from our experience with using the platform in these studies.

7.2 Future Research

Sections 3.7, 4.8, 5.5 and 6.7 mentioned various ideas for future research within the scope of this thesis. But the field of secure and privacy-friendly research data utilization presents large research opportunities that extend far beyond the scope of this thesis. In this section we want to provide an outlook on further areas that could be worthwhile exploring.

Methodological Work on Data Privacy

This potential research work involves the development of advanced techniques to enhance data privacy while preserving data utility. One can inspect existing approaches to improve factors such as performance, scalability, re-identification risks, or data quality. Additionally, investigating novel approaches such as DP and related semantic privacy models and their application in ML or synthetic data generation (see Section 2.7.6) can offer new insights into data protection strategies. Furthermore, focusing on domain-specific challenges and the customization of anonymization techniques for diverse data types, such as unstructured or multimedia data, can lead to custom solutions that address the requirements of different scenarios. By advancing research in data privacy techniques, researchers can contribute to the development of robust privacy-preserving tools that balance data privacy and data utility effectively in various contexts and applications.

Distributed Architectures

The application of data privacy techniques in distributed environments becomes more and more important with an increasing number of potentially collected data sources. While some architectures with a specific focus on anonymization and pseudonymization methods have been explored in this thesis, there is much room for future research. Researchers can investigate the design and implementation of distributed systems that incorporate privacy-preserving mechanisms to protect sensitive data across multiple parties. In addition to the application of basic techniques in distributed environments one can also look into advanced and novel methods such as SMPC, FL, or homomorphic encryption (HE) and their application to privacy-preserving data analysis. The intersection of privacy-preserving techniques and distributed architectures offers a wide range of research opportunities that can prove beneficial in a society where an increasing volume of personal data is stored across distributed databases.

Usable Privacy Technologies

Another line of potential research lies in exploring usability aspects of data privacy techniques. Investigating the user experience and interaction design of technical frameworks for these measures can enhance user acceptance and drive the adaptation of novel approaches. As an example, some problems hindering the practical implementation of privacy data techniques, in particular DP, are given in Section 2.7.7. Researchers can develop intuitive interfaces and educational tools that empower individuals to make informed decisions about adequate data privacy measures for their application in their specific usage context. Additionally, the provisioning of more automated solutions, in which, for example, privacy parameters are derived automatically or at least tool-supported, can prove beneficial for easier as well as more secure implementations. It is essential to support users in working with complex privacy techniques to increase the prevalence and correct use of these techniques.

Awareness and Education

Future research opportunities also exist in enhancing user awareness and education regarding the importance of privacy-preserving techniques such as anonymization and pseudonymization. By demonstrating practical examples of privacy breaches and potential risks associated with inadequate data protection measures (see Sections 2.6.8 and 2.8), non-experts can gain a better understanding of the importance of safeguarding personal information. Additionally, developing educational resources, such as tutorials or best practice recommendations, can empower individuals to make informed decisions about data privacy. By combining real-world examples with educational content, researchers can offer non-experts an accessible way to navigate technical complexities to increase privacy awareness and promote responsible data handling practices.

Interdisciplinary Collaboration

Another promising area of future research lies in the interdisciplinary collaboration between computer scientists, researchers in data privacy, legal scholars, data protection authorities, and policymakers to advance the field of data privacy techniques in accordance with privacy laws. By convening technical experts and legal professionals, studies can explore the intersection of privacy measures and regulatory frameworks to ensure compliance with privacy laws. One example for such studies is the deduction of formal requirements from laws covered in Section 2.9.3. However, it is just as important to investigate the implications of technical developments, such as improved data privacy techniques as well as new forms of attacks, to drive advancements in future privacy laws. Collaborative efforts can play an important role in addressing privacy challenges in a legally compliant fashion as well as driving the progression of regulatory frameworks respecting technical developments.

Real-World Applications

Finally, future research can look into the practical implementation of data privacy techniques in real-world applications across various domains. We have provided one example for a practical solution in the healthcare domain in Chapter 6. Exploring the integration of privacy-preserving techniques in other healthcare applications or further sensitive domains like finance can offer insights into the field-specific challenges and benefits of ensuring data privacy while maintaining data utility. This allows researchers to improve data privacy practices in real-world settings and to address the increasing data protection needs in our data-rich society.

7.3 Final Outlook

In this thesis we have have investigated the challenges and opportunities distributed environments pose to the application of data privacy methods, in particular pseudonymization and anonymization techniques. While our focus has been on medical research, it should not go unmentioned that the same or similar considerations apply to other fields of research based on personal data as well. As we have elaborated in Section 7.2, there a lot of further research opportunities focusing on various aspects of data privacy. The intersection of data science, regulatory frameworks, and technical privacy measures has significant consequences for the evolution of medical research. It is essential to balance the safeguarding of personal medical data with the growing data needs in the medical research domain. Interdisciplinary research and research-based regulatory efforts can help to achieve this balance. This final outlook encourages stakeholders to face the challenges and opportunities ahead, so that we can create a future in which research benefits from personal health data while simultaneously respecting the privacy of individuals.

A Appendix

A.1 Publications Related to the Thesis

The following publications are related to parts of this thesis.

[Pet+19] Tom Petersen, Maximilian Blochberger, Tobias Mueller, Hannes Federrath, and Christian-Alexander Behrendt. *Sichere und datenschutzgerechte Umsetzung medizinischer Register*. In: *Datenschutz und Datensicherheit - DuD* 43.8 (2019). DOI: 10.1007/s11623-019-1153-z.

[Pet20] Tom Petersen. *Datenschutzgerechte und mehrseitig sichere IT-Plattformen für die medizinische Forschung*. In: *SICHERHEIT 2020*. Gesellschaft für Informatik e.V., 2020. DOI: 10.18420/sicherheit2020_13.

[Zim+20] Ephraim Zimmer, Christian Burkert, Tom Petersen, and Hannes Federrath. *PEEPLL: Privacy-Enhanced Event Pseudonymisation with Limited Linkability*. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 2020. DOI: 10.1145/3341105.3375781.

A.2 Additional Publications

The following publications were created during the doctoral phase but are not directly related to the thesis.

[Beh+20] Christian-Alexander Behrendt, Thea Schwaneberg, Sandra Hischke, Tobias Müller, Tom Petersen, Ursula Marschall, Sebastian Debus, and Levente Kriston. *Data Privacy Compliant Validation of Health Insurance Claims Data: the IDOMENEO Approach*. In: *Das Gesundheitswesen* 82.S 02 (2020). DOI: 10.1055/a-0883-5098.

[BPF19] Maximilian Blochberger, Tom Petersen, and Hannes Federrath. *Mitigating Cryptographic Mistakes by Design*. In: *Mensch und Computer 2019*. 2019. DOI: 10.18420/muc2019-ws-302-02.

[PSF23] Tom Petersen, Joshua Stock, and Hannes Federrath. *Bedrohungsszenarien für Energieinfrastrukturen*. Working paper written as part of the project Norddeutsches Reallabor. 2023. URL: https://svs.informatik.uni-hamburg.de/publications/2023/2023-07-28-NRL-Whitepaper-UHH.pdf (visited on 10.5.2024).

[Sto+22a] Joshua Stock, Tom Petersen, Christian-Alexander Behrendt, Hannes Federrath, and Thea Kreutzburg. *Privatsphärefreundliches maschinelles Lernen – Teil 1: Grundlagen und Verfahren*. In: *Informatik Spektrum* 45.2 (2022). DOI: 10.1007/s00287-022-01438-3.

[Sto+22b] Joshua Stock, Tom Petersen, Christian-Alexander Behrendt, Hannes Federrath, and Thea Kreutzburg. *Privatsphärefreundliches maschinelles Lernen – Teil 2: Privatsphäreangriffe und Privacy-Preserving Machine Learning*. In: *Informatik Spektrum* 45.3 (2022). DOI: 10.1007/s00287-022-01440-9.

[Sto+23] Joshua Stock, Tom Petersen, David Zadim, and Hannes Federrath. *Abschlussbericht der Studie zur fachlichen Einschätzung und Prüfung des Potenzials von Federated-Learning-Algorithmen in der amtlichen Statistik*. Prepared for the German Federal Office of Statistics. 2023. URL: https://svs.informatik.uni-hamburg.de/publications/2023/2023-02-14-Abschlussbericht-FederatedLearning-final.pdf (visited on 10.5.2024).

Bibliography

We have tried to provide a DOI or URL for all bibliography entries, if possible.

[Aba+16] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. *Deep Learning with Differential Privacy*. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016. DOI: 10.1145/2976749.2978318.

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. *The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES*. In: *Topics in Cryptology — CT-RSA 2001*. 2001. DOI: 10.1007/3-540-45353-9_12.

[Agg+05] Gagan Aggarwal, Tomas Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. *Approximation Algorithms for k-Anonymity*. In: *Proceedings of the International Conference on Database Theory (ICDT 2005)*. 2005. URL: https://ptolemy.berkeley.edu/projects/truststc/pubs/100/k-anonymity-jopt.pdf.

[Agg05] Charu C. Aggarwal. *On K-Anonymity and the Curse of Dimensionality*. In: *Proceedings of the 31st International Conference on Very Large Data Bases*. 2005. URL: http://www.vldb.org/archives/website/2005/program/paper/fri/p901-aggarwal.pdf.

[Agg15] Charu C. Aggarwal. *Data Mining: the Textbook*. Springer, 2015. DOI: 10.1007/978-3-319-14142-8.

[AHK01] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. *On the Surprising Behavior of Distance Metrics in High Dimensional Space*. In: *Database Theory — ICDT 2001*. 2001. DOI: 10.1007/3-540-44503-X_27.

[AHS20] Jean-Philippe Aumasson, Adrian Hamelink, and Omer Shlomovits. *A Survey of ECDSA Threshold Signing*. Cryptology ePrint Archive, Paper 2020/1390. 2020. URL: https://eprint.iacr.org/2020/1390.

[Ala+21] Saruar Alam, Md. Kamrul Hasan, Sharif Neaz, Nazmul Hussain, Md. Faruk Hossain, and Tania Rahman. *Diabetes Mellitus: Insights from Epidemiology, Biochemistry, Risk Factors, Diagnosis, Complications and Comprehensive Management*. In: *Diabetology* 2.2 (2021). DOI: 10.3390/diabetology2020004.

[Alm+19] João Rafael Almeida, Olga Fajarda, Arnaldo Pereira, and José Luís Oliveira. *Strategies to Access Patient Clinical Data from Distributed Databases*. In: *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2019)*. 2019. DOI: 10.5220/0007576104660473.

[Alp16] Ravi Gulati Alpa Shah. *Privacy Preserving Data Mining: Techniques, Classification and Implications - A Survey*. In: *International Journal of Computer Applications* 137.12 (2016). DOI: 10.5120/ijca2016909006.

[Alt+21] Micah Altman, Aloni Cohen, Kobbi Nissim, and Alexandra Wood. *What a Hybrid Legal-Technical Analysis Teaches Us about Privacy Regulation: The Case of Singling out*. In: *Boston University Journal of Science and Technology Law* 27.1 (2021). URL: https://www.bu.edu/jostl/2021/08/06/vol-27-1-winter-2021/.

[Alv+12] Mário S. Alvim, Miguel E. Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. *Differential Privacy: On the Trade-Off between Utility and Information Leakage*. In: *Formal Aspects of Security and Trust*. 2012. DOI: 10.1007/978-3-642-29420-4_3.

[Aly+22] Abdelrahaman Aly, Benjamin Coenen, Kelong Cong, Karl Koch, Marcel Keller, Dragos Rotaru, Oliver Scherer, Peter Scholl, Nigel P. Smart, Titouan Tanguy, and Tim Wood. *SCALE-MAMBA*. 2022. URL: https://nigelsmart.github.io/SCALE/ (visited on 30.4.2024).

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. *Privacy-Preserving Data Mining*. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 2000. DOI: 10.1145/342009.335438.

[Ass+21] Samuel A. Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E. Tillman, Prashant Reddy, and Manuela Veloso. *Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls*. In: *Proceedings of the First ACM International Conference on AI in Finance*. 2021. DOI: 10.1145/3383455.3422554.

[Ate+06] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. *Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage*. In: *ACM Transactions on Information and System Security* 9.1 (2006). DOI: 10.1145/1127345.1127346.

[ATR14] Afonso Arriaga, Qiang Tang, and Peter Ryan. *Trapdoor Privacy in Asymmetric Searchable Encryption Schemes*. In: *Progress in Cryptology – AFRICACRYPT 2014*. 2014. DOI: 10.1007/978-3-319-06734-6_3.

[Ave+17] Brendan Avent, Aleksandra Korolova, David Zeber, Torgeir Hovden, and Benjamin Livshits. *BLENDER: Enabling Local Search with a Hybrid Differential Privacy Model*. In: *26th USENIX Security Symposium (USENIX Security 17)*. 2017. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/avent.

[AY08] Charu C. Aggarwal and Philip S. Yu. *A General Survey of Privacy-Preserving Data Mining Models and Algorithms*. In: *Privacy-Preserving Data Mining: Models and Algorithms*. Springer US, 2008. DOI: 10.1007/978-0-387-70992-5_2.

[Ayd+21] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. *Differentially Private Query Release Through Adaptive Projection*. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. 2021. URL: https://proceedings.mlr.press/v139/aydore21a.html.

[Bao+08] Feng Bao, Robert H. Deng, Xuhua Ding, and Yanjiang Yang. *Private Query on Encrypted Data in Multi-user Settings*. In: *Information Security Practice and Experience*. 2008. DOI: 10.1007/978-3-540-79104-1_6.

[Bar+15] Daniel Barth-Jones, Khaled El Emam, Jane Bambauer, Ann Cavoukian, and Bradley Malin. *Assessing data intrusion threats*. In: *Science* 348.6231 (2015). DOI: 10.1126/science.348.6231.194-b.

[Bas+13] Raef Bassily, Adam Groce, Jonathan Katz, and Adam Smith. *Coupled-Worlds Privacy: Exploiting Adversarial Uncertainty in Statistical Data Privacy*. In: *IEEE 54th Annual Symposium on Foundations of Computer Science*. 2013. DOI: 10.1109/FOCS.2013.54.

[BBS98] Matt Blaze, Gerrit Bleumer, and Martin Strauss. *Divertible protocols and atomic proxy cryptography*. In: *Advances in Cryptology – EUROCRYPT'98*. 1998. DOI: 10.1007/BFb0054122.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. *Keying Hash Functions for Message Authentication*. In: *Advances in Cryptology — CRYPTO '96*. 1996. DOI: 10.1007/3-540-68697-5_1.

[BD21] C.-A. Behrendt and E. S. Debus. *Vier Jahre IDOMENEO: Was hat Mozarts Oper der interdisziplinären Behandlung der peripheren arteriellen Verschlusskrankheit gebracht?* In: *Gefässchirurgie* 26.1 (2021). DOI: 10.1007/s00772-021-00745-5.

[Beh+17a] C.-A. Behrendt, M. Härter, L. Kriston, H. Federrath, U. Marschall, C. Straub, and E. S. Debus. *IDOMENEO - Ist die Versorgungsrealität in der Gefäßmedizin Leitlinien- und Versorgungsgerecht?* In: *Gefässchirurgie* 22.1 (2017). DOI: 10.1007/s00772-016-0234-7.

[Beh+17b] C.-A. Behrendt, H. Pridöhl, K. Schaar, H. Federrath, and E. S. Debus. *Klinische Register im 21. Jahrhundert*. In: *Der Chirurg* 88.11 (2017). DOI: 10.1007/s00104-017-0542-9.

[Beh+21] Christian-Alexander Behrendt, Hannes Federrath, Tobias Müller, and Tom Petersen. *Datenschutzkonzept für das GermanVasc-Register zur Behandlung der peripheren arteriellen Verschlusskrankheit (PAVK) in Deutschland*. Version 14. non-public. 2021.

[Beh+23] Christian-Alexander Behrendt, Markus Steinbauer, Irene Hinterseher, Livia Cotta, Farzin Adili, and Jörg Heckenkamp. *Moderne gefäßchirurgische Registerforschung*. In: *Gefässchirurgie* 28.5 (2023). DOI: 10.1007/s00772-023-01015-2.

[Ber+11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. *The KECCAK Reference*. 2011. URL: https://keccak.team/files/Keccak-reference-3.0.pdf (visited on 14.5.2024).

[Ber+22] Skye Berghel, Philip Bohannon, Damien Desfontaines, Charles Estes, Sam Haney, Luke Hartman, Michael Hay, Ashwin Machanavajjhala, Tom Magerlein, Gerome Miklau, Amritha Pai, William Sexton, and Ruchit Shrestha. *Tumult Analytics: a robust, easy-to-use, scalable, and expressive framework for differential privacy*. 2022. arXiv: 2212.04133 [cs.CR].

[Ber09] Daniel J. Bernstein. *Cryptography in NaCl*. 2009. URL: https://cr.yp.to/highspeed/naclcrypto-20090310.pdf (visited on 21.2.2024).

[Bey+20] Oya Beyan, Ananya Choudhury, Johan van Soest, Oliver Kohlbacher, Lukas Zimmermann, Holger Stenzhorn, Md. Rezaul Karim, Michel Dumontier, Stefan Decker, Luiz Olavo Bonino da Silva Santos, and Andre Dekker. *Distributed Analytics on Sensitive Medical Data: The Personal Health Train*. In: *Data Intelligence* 2.1-2 (2020). DOI: 10.1162/dint_a_00032.

[Bey+99] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. *When Is "Nearest Neighbor" Meaningful?* In: *Database Theory — ICDT'99*. 1999. DOI: 10.1007/3-540-49257-7_15.

[Bha+11] Raghav Bhaskar, Abhishek Bhowmick, Vipul Goyal, Srivatsan Laxman, and Abhradeep Thakurta. *Noiseless Database Privacy*. In: *Advances in Cryptology – ASIACRYPT 2011*. 2011. DOI: 10.1007/978-3-642-25385-0_12.

[Bit+17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. *Prochlo: Strong Privacy for Analytics in the Crowd*. In: *Proceedings of the 26th Symposium on Operating Systems Principles*. 2017. DOI: 10.1145/3132747.3132769.

[BJM05] S. G. Barwick, W. Jackson, and K. M. Martin. *Updating the parameters of a threshold scheme by minimal broadcast*. In: *IEEE Transactions on Information Theory* 51.2 (2005). DOI: 10.1109/TIT.2004.840857.

[BJN00] Dan Boneh, Antoine Joux, and Phong Q Nguyen. *Why textbook ElGamal and RSA encryption are insecure*. In: *Advances in Cryptology – ASIACRYPT 2000*. Vol. 1976. 2000. DOI: 10.1007/3-540-44448-3_3.

[BL20] Claire McKay Bowen and Fang Liu. *Comparative Study of Differentially Private Data Synthesis Methods*. In: *Statistical Science* 35.2 (2020). DOI: 10.1214/19-sts742.

[Bla+93] Bob Blakley, G. R. Blakley, A. H. Chan, and J. L. Massey. *Threshold Schemes with Disenrollment*. In: *Advances in Cryptology – CRYPTO' 92*. 1993. DOI: 10.1007/3-540-48071-4_38.

[Blo70] Burton H Bloom. *Space/time trade-offs in hash coding with allowable errors*. In: *Communications of the ACM* 13.7 (1970). DOI: 10.1145/362686.362692.

[BLR13] Avrim Blum, Katrina Ligett, and Aaron Roth. *A Learning Theory Approach to Noninteractive Database Privacy*. In: *Journal of the ACM* 60.2 (2013). DOI: 10.1145/2450142.2450148.

[Blu+94] C. Blundo, A. Cresti, A. De Santis, and U. Vaccaro. *Fully Dynamic Secret Sharing Schemes*. In: *Advances in Cryptology — CRYPTO' 93*. 1994. DOI: 10.1007/3-540-48329-2_10.

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. *The Round Complexity of Secure Protocols*. In: *Proceedings of the 22nd annual ACM symposium on Theory of Computing (STOC'90)*. 1990. DOI: 10.1145/100216.100287.

[BMS13] Jane Bambauer, Krishnamurty Muralidhar, and Rathindra Sarathy. *Fool's Gold: An Illustrated Critique of Differential Privacy*. In: *Vanderbilt Journal of Entertainment and Technology Law* 16.4 (2013). URL: https://heinonline.org/HOL/P?h=hein.journals/vanep16&i=734.

[Bol02] Alexandra Boldyreva. *Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme*. In: *Public Key Cryptography (PKC 2003)*. 2002.

[Bon+04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. *Public Key Encryption with Keyword Search*. In: *Advances in Cryptology – EUROCRYPT 2004*. 2004. DOI: 10.1007/978-3-540-24676-3_30.

[Bös+14] Christoph Bösch, Pieter Hartel, Willem Jonker, and Andreas Peter. *A survey of provably secure searchable encryption*. In: *ACM Computing Surveys (CSUR)* 47.2 (2014). DOI: 10.1145/2636328.

[BP21] Marcus Brubaker and Simon Prince. *Differential Privacy I: Introduction*. 2021. URL: https://www.borealisai.com/en/blog/tutorial-12-differential-privacy-i-introduction/ (visited on 14.7.2022).

[BQS24] BQS. *Registerdatenbank der medizinischen Register in Deutschland*. 2024. URL: https://registersuche.bqs.de/ (visited on 1.2.2024).

[Bra+22] Lennart Braun, Daniel Demmler, Thomas Schneider, and Oleksandr Tkachenko. *MOTION – A Framework for Mixed-Protocol Multi-Party Computation*. In: *Transactions on Privacy and Security* 25.2 (2022). DOI: 10.1145/3490390.

[BS08] Justin Brickell and Vitaly Shmatikov. *The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing*. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008. DOI: 10.1145/1401890.1401904.

[BS20] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. 2020. URL: https://toc.cryptobook.us (visited on 30.11.2020).

[BS22] Lorina Buhr and Silke Schicktanz. *Individual benefits and collective challenges: Experts' views on data-driven approaches in medical research and healthcare in the German context*. In: *Big Data & Society* 9.1 (2022). DOI: 10.1177/20539517221092653.

[BSI23] BSI. *IT-Grundschutz-Kompendium – Werkzeug für Informationssicherheit*. 2023. URL: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompendium/it-grundschutz-kompendium_node.html (visited on 28.3.2024).

[Bul+17] Brooke Bullek, Stephanie Garboski, Darakhshan J. Mir, and Evan M. Peck. *Towards Understanding Differential Privacy: When Do People Trust Randomized Response Technique?* In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017. DOI: 10.1145/3025453.3025698.

[BW18] Borja Balle and Yu-Xiang Wang. *Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising*. 2018. arXiv: 1805.06530 [cs.LG].

[Byu+06] Ji-Won Byun, Yonglak Sohn, Elisa Bertino, and Ninghui Li. *Secure Anonymization for Incremental Datasets*. In: *Secure Data Management (SDM 2006)*. 2006. DOI: 10.1007/11844662_4.

[BZH06] Michael Barbaro, Tom Zeller, and Saul Hansell. *A face is exposed for AOL searcher no. 4417749*. In: *New York Times* (2006). Issue dated 09.08.2006. URL: https://www.nytimes.com/2006/08/09/technology/09aol.html.

[CA15] Julia Carnevale and Alan Ashworth. *Assessing the Significance of BRCA1 and BRCA2 Mutations in Pancreatic Cancer*. In: *Journal of Clinical Oncology* 33.28 (2015). DOI: 10.1200/JCO.2015.61.6961.

[Car+23] Tânia Carvalho, Nuno Moniz, Pedro Faria, and Luís Antunes. *Survey on Privacy-Preserving Techniques for Microdata Publication*. In: *ACM Computing Surveys* 55.14s (2023). DOI: 10.1145/3588765.

[CC14] Ann Cavoukian and Daniel Castro. *Big Data and Innovation, Setting the Record Straight: De-identification Does Work*. 2014. URL: https://www2.itif.org/2014-big-data-deidentification.pdf (visited on 14.5.2023).

[CH07] Ran Canetti and Susan Hohenberger. *Chosen-Ciphertext Secure Proxy Re-encryption*. In: *Proceedings of the 14th ACM conference on Computer and communications security (CCS'07)*. 2007. DOI: 10.1145/1315245.1315269.

[Che+09] Bee-Chung Chen, Daniel Kifer, Kristen LeFevre, Ashwin Machanavajjhala, et al. *Privacy-preserving data publishing*. In: *Foundations and Trends® in Databases* 2.1–2 (2009). DOI: 10.1561/1900000008.

[Che+15] Feng Chen, Noman Mohammed, Shuang Wang, Wenbo He, Samuel Cheng, and Xiaoqian Jiang. *Cloud-Assisted Distributed Private Data Sharing*. In: *Conference on Bioinformatics, Computational Biology and Health Informatics*. 2015. DOI: 10.1145/2808719.2808740.

[Chr+17] Peter Christen, Rainer Schnell, Dinusha Vatsalan, and Thilina Ranbaduge. *Efficient Cryptanalysis of Bloom Filters for Privacy-Preserving Record Linkage*. In: *Advances in Knowledge Discovery and Data Mining*. 2017. DOI: 10.1007/978-3-319-57454-7_49.

[Chu+22] Bilal Chughtai, Sirikan Rojanasarot, Kurt Neeser, Dmitry Gultyaev, Shuai Fu, Samir K. Bhattacharyya, Ahmad M. El-Arabi, Ben J. Cutone, and Kevin T. McVary. *A comprehensive analysis of clinical, quality of life, and cost-effectiveness outcomes of key treatment options for benign prostatic hyperplasia*. In: *PLOS ONE* 17.4 (2022). DOI: 10.1371/journal.pone.0266824.

[CKR21] Rachel Cummings, Gabriel Kaptchuk, and Elissa M. Redmiles. *"I Need a Better Description": An Investigation Into User Expectations For Differential Privacy*. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021. DOI: 10.1145/3460120.3485252.

[Cla94] Kenneth L. Clarkson. *An Algorithm for Approximate Closest-Point Queries*. In: *Proceedings of the Tenth Annual Symposium on Computational Geometry*. 1994. DOI: 10.1145/177424.177609.

[Cli+02] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y. Zhu. *Tools for privacy preserving distributed data mining*. In: *ACM SIGKDD Explorations Newsletter* 4.2 (2002). DOI: 10.1145/772862.772867.

[CM05] Yan-Cheng Chang and Michael Mitzenmacher. *Privacy Preserving Keyword Searches on Remote Encrypted Data*. In: *Applied Cryptography and Network Security*. 2005. DOI: 10.1007/11496137_30.

[CMV21] Mariana Cunha, Ricardo Mendes, and João P. Vilela. *A survey of privacy-preserving mechanisms for heterogeneous data types*. In: *Computer Science Review* 41 (2021). DOI: https://doi.org/10.1016/j.cosrev.2021.100403.

[CN20] Aloni Cohen and Kobbi Nissim. *Towards formalizing the GDPR's notion of singling out*. In: *Proceedings of the National Academy of Sciences* 117.15 (2020). DOI: 10.1073/pnas.1914598117.

[Coh22] Aloni Cohen. *Attacks on Deidentification's Defenses*. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/cohen.

[Con+05] P. Contiero, A. Tittarelli, G. Tagliabue, A. Maghini, S. Fabiano, P. Crosignani, and R. Tessandori. *The EpiLink record linkage software*. In: *Methods of Information in Medicine* 44.01 (2005). DOI: 10.1055/s-0038-1633924.

[Cor+10] Graham Cormode, Divesh Srivastava, Ninghui Li, and Tiancheng Li. *Minimizing Minimality and Maximizing Utility: Analyzing Method-Based Attacks on Anonymized Data*. In: *Proceedings of the VLDB Endowment* 3.1–2 (2010). DOI: 10.14778/1920841.1920972.

[Cor+18] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. *Privacy at Scale: Local Differential Privacy in Practice*. In: *Proceedings of the 2018 International Conference on Management of Data*. 2018. DOI: 10.1145/3183713.3197390.

[Cor11] Graham Cormode. *Personal Privacy vs Population Privacy: Learning to Attack Anonymization*. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2011. DOI: 10.1145/2020408.2020598.

[CRT17] Chris Culnane, Benjamin I. P. Rubinstein, and Vanessa Teague. *Health Data in an Open World*. 2017. arXiv: 1712.05627 [cs.CY].

[CT13] C. Clifton and T. Tassa. *On syntactic anonymity and differential privacy*. In: *International Conference on Data Engineering Workshops*. 2013. DOI: 10.1109/ICDEW.2013.6547433.

[Cur+06] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. *Searchable symmetric encryption: improved definitions and efficient constructions*. In: *Proceedings of the 13th ACM conference on Computer and communications security*. 2006. DOI: 10.1145/1180405.1180417.

[CV15] Francis S. Collins and Harold Varmus. *A New Initiative on Precision Medicine*. In: *New England Journal of Medicine* 372.9 (2015). DOI: 10.1056/NEJMp1500523.

[Dal77] Tore Dalenius. *Towards a methodology for statistical disclosure control*. In: *Statistisk tidskrift* 15 (1977). URL: https://hdl.handle.net/1813/111303.

[Dal86] Tore Dalenius. *Finding a Needle In a Haystack or Identifying Anonymous Census Records*. In: *Journal of Official Statistics* 2.3 (1986). URL: https://www.proquest.com/scholarly-journals/finding-needle-haystack-identifying-anonymous/docview/1266806751/se-2.

[Dam+12] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. *Multiparty Computation from Somewhat Homomorphic Encryption*. In: *Advances in Cryptology – CRYPTO 2012*. 2012. DOI: 10.1007/978-3-642-32009-5_38.

[De +12] Sabrina De Capitani Di Vimercati, Sara Foresti, Giovanni Livraga, and Pierangela Samarati. *Data Privacy: Definitions and Techniques*. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 20.06 (2012). DOI: 10.1142/S0218488512400247.

[De +13] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. *Unique in the crowd: The privacy bounds of human mobility*. In: *Scientific reports* 3.1 (2013). DOI: 10.1038/srep01376.

[DE13] Fida K. Dankar and Khaled El Emam. *Practicing differential privacy in health care: A review*. In: *Transactions on Data Privacy* 6.1 (2013). URL: https://dl.acm.org/doi/abs/10.5555/2612156.2612159.

[Dee24] Google DeepMind. *AlphaZero and MuZero*. 2024. URL: https://deepmind.google/technologies/alphazero-and-muzero/ (visited on 29.8.2024).

[Dem+18] Levent Demir, Amrit Kumar, Mathieu Cunche, and Cédric Lauradoux. *The Pitfalls of Hashing for Privacy*. In: *IEEE Communications Surveys & Tutorials* 20.1 (2018). DOI: 10.1109/COMST.2017.2747598.

[Des18] Damien Desfontaines. *Differential privacy in (a bit) more detail*. https://desfontain.es/privacy/differential-privacy-in-more-detail.html. Ted is writing things (personal blog). 2018 (visited on. 1.7.2022).

[Des20] Damien Desfontaines. *The privacy loss random variable*. https://desfontain.es/privacy/privacy-loss-random-variable.html. Ted is writing things (personal blog). 2020 (visited on. 16.9.2022).

[Des21] Damien Desfontaines. *Don't worry, your data's noisy*. https://desfontain.es/privacy/noisy-data.html. Ted is writing things (personal blog). 2021.

[DF90] Yvo Desmedt and Yair Frankel. *Threshold cryptosystems*. In: *Advances in Cryptology — CRYPTO'89*. 1990. DOI: 10.1007/0-387-34805-0_28.

[DI20] German Federal Commissioner for Data Protection and Freedom of Information. *Positionspapier zur Anonymisierung unter der DSGVO unter besonderer Berücksichtigung der TK-Branche*. 2020. URL: https://www.bfdi.bund.de/DE/DerBfDI/Konsultationsverfahren/Anonymisierung-TK/Anonymisierung-TK_node.html (visited on 14.7.2022).

[Dic+23] Travis Dick, Cynthia Dwork, Michael Kearns, Terrance Liu, Aaron Roth, Giuseppe Vietri, and Zhiwei Steven Wu. *Confidence-ranked reconstruction of census microdata from published statistics*. In: *Proceedings of the National Academy of Sciences* 120.8 (2023). DOI: 10.1073/pnas.2218605120.

[Dif23] DifferentialPrivacy.org. *Differential Privacy: Resources*. 2023. URL: https://differentialprivacy.org/resources/ (visited on 30.9.2023).

[DJ97] Yvo Desmedt and Sushil Jajodia. *Redistributing secret shares to new access structures and its applications*. Tech. rep. ISSE TR-97-01, George Mason University, 1997. URL: https://csis.gmu.edu/jajodia/desmedt25-7-1997.pdf.

[DKM19] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. *Differential Privacy in Practice: Expose your Epsilons!* In: *Journal of Privacy and Confidentiality* 9.2 (2019). DOI: 10.29012/jpc.689.

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. *Collecting Telemetry Data Privately*. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/253614bbac999b38b5b60cae531c4969-Paper.pdf.

[DL89] George Duncan and Diane Lambert. *The risk of disclosure for microdata*. In: *Journal of Business & Economic Statistics* 7.2 (1989). DOI: 10.1080/07350015.1989.10509729.

[Dom08] Josep Domingo-Ferrer. *A Survey of Inference Control Methods for Privacy-Preserving Data Mining*. In: *Privacy-Preserving Data Mining: Models and Algorithms*. Springer US, 2008. DOI: 10.1007/978-0-387-70992-5_3.

[Dou+16] Marie Douriez, Harish Doraiswamy, Juliana Freire, and Cláudio T. Silva. *Anonymizing NYC Taxi Data: Does It Matter?* In: *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 2016. DOI: 10.1109/DSAA.2016.21.

[DP20] Damien Desfontaines and Balázs Pejó. *SoK: Differential Privacies*. In: *Proceedings on Privacy Enhancing Technologies* 2020.2 (2020). DOI: 10.2478/popets-2020-0028.

[DR+14] Cynthia Dwork, Aaron Roth, et al. *The algorithmic foundations of differential privacy*. In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014). DOI: 10.1561/0400000042.

[DS15] Josep Domingo-Ferrer and Jordi Soria-Comas. *From t-closeness to differential privacy and vice versa in data anonymization*. In: *Knowledge-Based Systems* 74 (2015). DOI: https://doi.org/10.1016/j.knosys.2014.11.011.

[DSB21] Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia. *The Limits of Differential Privacy (and Its Misuse in Data Release and Machine Learning)*. In: *Communications of the ACM* 64.7 (2021). DOI: 10.1145/3433638.

[DSZ15] Daniel Demmler, Thomas Schneider, and Michael Zohner. *ABY – A Framework for Efficient Mixed-Protocol Secure Two-Party Computation*. In: *Proceedings of the 2015 Network and Distributed System Security Symposium (NDSS'15)*. 2015. URL: https://www.ndss-symposium.org/ndss2015/ndss-2015-programme/aby-framework-efficient-mixed-protocol-secure-two-party-computation/.

[Dwo+06a] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. *Our Data, Ourselves: Privacy Via Distributed Noise Generation*. In: *Advances in Cryptology – EUROCRYPT 2006*. 2006. DOI: 10.1007/11761679_29.

[Dwo+06b] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. *Calibrating Noise to Sensitivity in Private Data Analysis*. In: *Theory of Cryptography*. 2006. DOI: 10.1007/11681878_14.

[Dwo06] Cynthia Dwork. *Differential Privacy*. In: *International Colloquium on Automata, Languages, and Programming (ICALP)*. 2006. DOI: 10.1007/11787006_1.

[EÁ14] Khaled El Emam and Cecilia Álvarez. *A critical appraisal of the Article 29 Working Party Opinion 05/2014 on data anonymization techniques*. In: *International Data Privacy Law* 5.1 (2014). DOI: 10.1093/idpl/ipu033.

[Eke+22] Emelie Ekenstedt, Lawrence Ong, Yucheng Liu, Sarah Johnson, Phee Lep Yeoh, and Joerg Kliewer. *When Differential Privacy Implies Syntactic Privacy*. In: *IEEE Transactions on Information Forensics and Security* 17 (2022). DOI: 10.1109/TIFS.2022.3177953.

[EKR18] David Evans, Vladimir Kolesnikov, and Mike Rosulek. *A pragmatic introduction to secure multi-party computation*. In: *Foundations and Trends in Privacy and Security* 2.2-3 (2018). DOI: 10.1561/3300000019.

[El +09] Khaled El Emam, Fida Kamal Dankar, Romeo Issa, Elizabeth Jonker, Daniel Amyot, Elise Cogo, Jean-Pierre Corriveau, Mark Walker, Sadrul Chowdhury, Regis Vaillancourt, et al. *A globally optimal k-anonymity method for the de-identification of health data*. In: *Journal of the American Medical Informatics Association* 16.5 (2009). DOI: 10.1197/jamia.M3144.

[El 13] Khaled El Emam. *Guide to the de-identification of personal health information*. CRC Press, 2013. DOI: 10.1201/b14764.

[Elg85] Taher Elgamal. *A public key cryptosystem and a signature scheme based on discrete logarithms*. In: *IEEE Transactions on Information Theory* 31.4 (1985). DOI: 10.1109/TIT.1985.1057074.

[ENI18] ENISA. *Recommendations on shaping technology according to GDPR provisions - An overview on data pseudonymisation*. 2018. URL: https://www.enisa.europa.eu/publications/recommendations-on-shaping-technology-according-to-gdpr-provisions (visited on 30.1.2023).

[ENI19] ENISA. *Pseudonymisation techniques and best practices*. 2019. URL: https://www.enisa.europa.eu/publications/pseudonymisation-techniques-and-best-practices (visited on 30.1.2023).

[ENI21] ENISA. *Data Pseudonymisation: Advanced Techniques and Use Cases*. 2021. URL: https://www.enisa.europa.eu/publications/data-pseudonymisation-advanced-techniques-and-use-cases (visited on 30.1.2023).

[ENI22] ENISA. *Deploying Pseudonymization Techniques*. 2022. URL: https://www.enisa.europa.eu/publications/deploying-pseudonymisation-techniques (visited on 30.1.2023).

[epi] epic.org. *U.S. Privacy Laws*. URL: https://epic.org/issues/privacy-laws/united-states/ (visited on 7.6.2023).

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. *RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response*. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014. DOI: 10.1145/2660267.2660348.

[Far23] Harshwardhan Fartale. *A Survey of Differential Privacy Frameworks*. OpenMined. 2023. URL: https://blog.openmined.org/a-survey-of-differential-privacy-frameworks/ (visited on 25.11.2023).

[Fel87] Paul Feldman. *A practical scheme for non-interactive verifiable secret sharing*. In: *28th Annual Symposium on Foundations of Computer Science (SFCS 1987)*. 1987. DOI: 10.1109/SFCS.1987.4.

[FI19] Sam Fletcher and Md. Zahidul Islam. *Decision Tree Classification with Differential Privacy: A Survey*. In: *ACM Computing Surveys* 52.4 (2019). DOI: 10.1145/3337064.

[FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. *Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures*. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 2015. DOI: 10.1145/2810103.2813677.

[For21] MII Data Protection Task Force. *Übergreifendes Datenschutzkonzept der Medizininformatik-Initiative*. 2021. URL: https://www.medizininformatik-initiative.de/en/data-protection-concept (visited on 22.2.2024).

[Fra+22] Daniel Franzen, Saskia Nuñez von Voigt, Peter Sörries, Florian Tschorsch, and Claudia Müller-Birn. *Am I Private and If So, how Many?* In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2022. DOI: 10.1145/3548606.3560693.

[Fra24] Goethe University Frankfurt. *OSSE – Open Source Registry System for Rare Diseases*. 2024. URL: https://en.osse-register.de/en/ (visited on 24.1.2024).

[FS10] Arik Friedman and Assaf Schuster. *Data Mining with Differential Privacy*. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2010. DOI: 10.1145/1835804.1835868.

[Fun+08] Benjamin C. M. Fung, Ke Wang, Ada Wai-Chee Fu, and Jian Pei. *Anonymity for Continuous Data Publishing*. In: *Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology*. 2008. DOI: 10.1145/1353343.1353378.

[Fun+10a] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. *Privacy-Preserving Data Publishing: A Survey of Recent Developments*. In: *ACM Computing Surveys* 42.4 (2010). DOI: 10.1145/1749603.1749605.

[Fun+10b] Benjamin CM Fung, Ke Wang, Ada Wai-Chee Fu, and S Yu Philip. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Chapman and Hall/CRC, 2010. DOI: 10.1201/9781420091502.

[FV22] Alvaro Figueira and Bruno Vaz. *Survey on Synthetic Data Generation, Evaluation Methods and GANs*. In: *Mathematics* 10.15 (2022). DOI: 10.3390/math10152733.

[FWP07] Benjamin CM Fung, Ke Wang, and S Yu Philip. *Anonymizing classification data for privacy preservation*. In: *IEEE Transactions on Knowledge and Data engineering* 19.5 (2007). DOI: 10.1109/TKDE.2007.1015.

[FWY05] B.C.M. Fung, K. Wang, and P.S. Yu. *Top-down specialization for information and privacy preservation*. In: *21st International Conference on Data Engineering (ICDE'05)*. 2005. DOI: 10.1109/ICDE.2005.143.

[FZ08] Keith B. Frikken and Yihua Zhang. *Yet Another Privacy Metric for Publishing Micro-Data*. In: *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*. 2008. DOI: 10.1145/1456403.1456423.

[Gab+18] Marco Gaboardi, James Honaker, Gary King, Jack Murtagh, Kobbi Nissim, Jonathan Ullman, and Salil Vadhan. *PSI: a Private data Sharing Interface*. 2018. arXiv: 1609.04340 [cs.CR].

[Gar+23] Gonzalo M Garrido, Xiaoyuan Liu, Florian Matthes, and Dawn Song. *Lessons Learned: Surveying the Practicality of Differential Privacy in the Industry*. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (2023). DOI: 10.56553/popets-2023-0045.

[Gar14] Simson Garfinkel. *NISTIR 8053: De-Identification of Personal Information*. NIST. 2014. URL: https://csrc.nist.gov/publications/detail/nistir/8053/final (visited on 4.6.2023).

[Gen+99] Rosario Gennaro, Stanisław Jarecki, Hugo Krawczyk, and Tal Rabin. *Secure distributed key generation for discrete-log based cryptosystems*. In: *Advances in Cryptology — EUROCRYPT '99*. Vol. 1592. 1999. DOI: 10.1007/3-540-48910-X_21.

[Ghi+07] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. *Fast data anonymization with low information loss*. In: *Proceedings of the 33rd international conference on Very large data bases*. 2007. DOI: http://www.vldb.org/conf/2007/papers/research/p758-ghinita.pdf.

[GHV20] Marco Gaboardi, Michael Hay, and Salil Vadhan. *Programming Framework for OpenDP*. 2020. URL: https://projects.iq.harvard.edu/files/opendifferentialprivacy/files/opendp_programming_framework_11may2020_1_01.pdf (visited on 14.11.2023).

[Gko+21] Aris Gkoulalas-Divanis, Dinusha Vatsalan, Dimitrios Karapiperis, and Murat Kantarcioglu. *Modern Privacy-Preserving Record Linkage Techniques: An Overview*. In: *IEEE Transactions on Information Forensics and Security* 16 (2021). DOI: 10.1109/TIFS.2021.3114026.

[GKS08] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. *Composition Attacks and Auxiliary Information in Data Privacy*. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008. DOI: 10.1145/1401890.1401926.

[GKW98] J. M. Gouweleeuw, P. Kooiman, and P-P Wolf. *Post Randomisation for Statistical Disclosure Control: Theory and Implementation*. In: *Journal of Official Statistics* 14.4 (1998). URL: https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/post-randomisation-for-statistical-disclosure-control-theory-and-implementation.pdf.

[GLS14] Aris Gkoulalas-Divanis, Grigorios Loukides, and Jimeng Sun. *Publishing data from electronic health records while preserving privacy: A survey of algorithms*. In: *Journal of Biomedical Informatics* 50 (2014). Special Issue on Informatics Methods in Medical Privacy. DOI: https://doi.org/10.1016/j.jbi.2014.06.002.

[GM84] Shafi Goldwasser and Silvio Micali. *Probabilistic encryption*. In: *Journal of Computer and System Sciences* 28.2 (1984). DOI: https://doi.org/10.1016/0022-0000(84)90070-9.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. *How to play ANY mental game*. In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC '87)*. 1987. DOI: 10.1145/28395.28420.

[Goh03] Eu-Jin Goh. *Secure Indexes*. Cryptology ePrint Archive, Paper 2003/216. 2003. URL: https://eprint.iacr.org/2003/216.

[Gre+69] Bernard G. Greenberg, Abdel-Latif A. Abul-Ela, Walt R. Simmons, and Daniel G. Horvitz. *The Unrelated Question Randomized Response Model: Theoretical Framework*. In: *Journal of the American Statistical Association* 64.326 (1969). DOI: 10.1080/01621459.1969.10500991.

[GS19] Alexander Gabel and Ina Schiering. *Privacy Patterns for Pseudonymity*. In: *Privacy and Identity Management. Fairness, Accountability, and Transparency in the Age of Big Data.* Springer International Publishing, 2019. DOI: 10.1007/978-3-030-16744-8_11.

[GTC22] Kaycee E. Glattke, Sailesh V. Tummala, and Anikar Chhabra. *Anterior Cruciate Ligament Reconstruction Recovery and Rehabilitation: A Systematic Review*. In: *The Journal of Bone and Joint Surgery* 104.8 (2022). DOI: 10.2106/JBJS.21.00688.

[Gue+20] Miguel Guevara, Damien Desfontaines, Jim Waldo, and Terry Coatta. *Differential Privacy: The Pursuit of Protections by Default: A Discussion with Miguel Guevara, Damien Desfontaines, Jim Waldo, and Terry Coatta*. In: *Queue* 18.5 (2020). DOI: 10.1145/3434571.3439229.

[GXF14] Slawomir Goryczka, Li Xiong, and Benjamin C. M. Fung. *m-Privacy for Collaborative Data Publishing*. In: *IEEE Transactions on Knowledge and Data Engineering* 26.10 (2014). DOI: 10.1109/TKDE.2013.18.

[Har15] Moritz Hardt. *Towards practicing differential privacy*. 2015. URL: http://blog.mrtz.org/2015/03/13/practicing-differential-privacy.html (visited on 13.6.2023).

[Hay+16] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, Dan Zhang, and George Bissias. *Exploring Privacy-Accuracy Tradeoffs Using DPComp*. In: *Proceedings of the 2016 International Conference on Management of Data*. 2016. DOI: 10.1145/2882903.2899387.

[HB20] Patrik Hummel and Matthias Braun. *Just data? Solidarity and justice in data-driven medicine*. In: *Life Sciences, Society and Policy* 16.1 (2020). DOI: 10.1186/s40504-020-00101-7.

[HBN11] Yeye He, Siddharth Barman, and Jeffrey F. Naughton. *Preventing equivalence attacks in updated, anonymized data*. In: *IEEE 27th International Conference on Data Engineering*. 2011. DOI: 10.1109/ICDE.2011.5767924.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Her+22] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. *Synthetic data generation for tabular health records: A systematic review*. In: *Neurocomputing* 493 (2022). DOI: https://doi.org/10.1016/j.neucom.2022.04.053.

[Her+95] Amir Herzberg, Stanisław Jarecki, Hugo Krawczyk, and Moti Yung. *Proactive Secret Sharing Or: How to Cope With Perpetual Leakage*. In: *Advances in Cryptology — CRYPT0' 95*. 1995. DOI: 10.1007/3-540-44750-4_27.

[Her23] Jeremy Herbst. *Pseudonymtreuhänder mithilfe von Searchable Encryption*. Bachelor's thesis. Universität Hamburg, 2023.

[HLM12] Moritz Hardt, Katrina Ligett, and Frank Mcsherry. *A Simple and Practical Algorithm for Differentially Private Data Release*. In: 25 (2012). URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/208e43f0e45c4c78cafadb83d2888cb6-Paper.pdf.

[Hol+19] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. *Diffprivlib: The IBM Differential Privacy Library*. 2019. arXiv: 1907.02444 [cs.CR].

[Hsu+14] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C. Pierce, and Aaron Roth. *Differential Privacy: An Economic Method for Choosing Epsilon*. In: *IEEE 27th Computer Security Foundations Symposium*. 2014. DOI: 10.1109/CSF.2014.35.

[Hu+24] Y. Hu, F. Wu, Q. Li, Y. Long, G. Garrido, C. Ge, B. Ding, D. Forsyth, B. Li, and D. Song. *SoK: Privacy-Preserving Data Synthesis*. In: *2024 IEEE Symposium on Security and Privacy (SP)*. 2024. DOI: 10.1109/SP54263.2024.00002.

[Hun+12] Anco Hundepool, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter-Paul De Wolf. *Statistical disclosure control*. Wiley New York, 2012. DOI: 10.1002/9781118348239.

[ico22] ICO. *ICO call for views: Anonymisation, pseudonymisation and privacy enhancing technologies guidance*. 2022. URL: https://ico.org.uk/about-the-ico/ico-and-stakeholder-consultations/ico-call-for-views-anonymisation-pseudonymisation-and-privacy-enhancing-technologies-guidance/ (visited on 25.5.2023).

[ID03] Anca-Andreea Ivan and Yevgeniy Dodis. *Proxy Cryptography Revisited*. In: *Proceedings of the 2003 Symposium on Network and Distributed System Security (NDSS'03)*. 2003. URL: https://www.ndss-symposium.org/ndss2003/proxy-cryptography-revisited/.

[IM98] Piotr Indyk and Rajeev Motwani. *Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality*. In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. 1998. DOI: 10.1145/276698.276876.

[Ish+03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. *Extending Oblivious Transfers Efficiently*. In: *Advances in Cryptology - CRYPTO 2003*. 2003. DOI: 10.1007/978-3-540-45146-4_9.

[ISN89] Mitsuru Ito, Akira Saito, and Takao Nishizeki. *Secret sharing scheme realizing general access structure*. In: *Electronics and Communications in Japan Part III* 72.9 (1989). DOI: 10.1002/ecjc.4430720906.

[Jak99] Markus Jakobsson. *On Quorum Controlled Asymmetric Proxy Re-encryption*. In: *Public Key Cryptography*. 1999. DOI: 10.1007/3-540-49162-7_9.

[JC06] Wei Jiang and Chris Clifton. *A secure distributed framework for achieving k-anonymity*. In: *The VLDB Journal* 15.4 (2006). DOI: 10.1007/s00778-006-0008-z.

[JE19] Bargav Jayaraman and David Evans. *Evaluating Differentially Private Machine Learning in Practice*. In: *28th USENIX Security Symposium (USENIX Security 19)*. 2019. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman.

[Joh+20] Noah Johnson, Joseph P. Near, Joseph M. Hellerstein, and Dawn Song. *Chorus: a Programming Framework for Building Scalable Differential Privacy Mechanisms*. In: *IEEE European Symposium on Security and Privacy (EuroS&P)*. 2020. DOI: 10.1109/EuroSP48549.2020.00041.

[Joh+21] Mark F. St. John, Grit Denker, Peeter Laud, Karsten Martiny, Alisa Pankova, and Dusko Pavlovic. *Decision Support for Sharing Data using Differential Privacy*. In: *IEEE Symposium on Visualization for Cyber Security (VizSec)*. 2021. DOI: 10.1109/VizSec53666.2021.00008.

[Jor+22] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. *Synthetic Data – what, why and how?* 2022. arXiv: 2205.03257 [cs.LG].

[JX08] Pawel Jurczyk and Li Xiong. *Towards privacy-preserving integration of distributed heterogeneous data*. In: *Proceedings of the 2nd PhD Workshop on Information and Knowledge Management*. 2008. DOI: 10.1145/1458550.1458562.

[JX09] Pawel Jurczyk and Li Xiong. *Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers*. In: *IFIP Annual Conference on Data and Applications Security and Privacy*. 2009. DOI: 10.1007/978-3-642-03007-9_13.

[JYC15] Zach Jorgensen, Ting Yu, and Graham Cormode. *Conservative or liberal? Personalized differential privacy*. In: *IEEE 31st International Conference on Data Engineering*. 2015. DOI: 10.1109/ICDE.2015.7113353.

[JZG22] Xue Jiang, Xuebing Zhou, and Jens Grossklags. *Privacy-Preserving High-dimensional Data Collection with Federated Generative Autoencoder*. In: *Proceedings on Privacy Enhancing Technologies* 2022.1 (2022). DOI: 10.2478/popets-2022-0024.

[KAF22] Farzaneh Karegar, Ala Sarah Alaqra, and Simone Fischer-Hübner. *Exploring User-Suitable Metaphors for Differentially Private Data Analyses*. In: *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*. 2022. URL: https://www.usenix.org/conference/soups2022/presentation/karegar.

[Kai+21] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. *Advances and open problems in federated learning*. In: *Foundations and Trends® in Machine Learning* 14.1–2 (2021). DOI: 10.1561/2200000083.

[Kar+03] H. Kargupta, S. Datta, Q. Wang, and Krishnamoorthy Sivakumar. *On the privacy preserving properties of random data perturbation techniques*. In: *Third IEEE International Conference on Data Mining*. 2003. DOI: 10.1109/ICDM.2003.1250908.

[Kas+11] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. *What Can We Learn Privately?* In: *SIAM Journal on Computing* 40.3 (2011). DOI: 10.1137/090756090.

[KBC97] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104. 1997. DOI: 10.17487/RFC2104.

[Kel20] Marcel Keller. *MP-SPDZ: A Versatile Framework for Multi-Party Computation*. In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020. DOI: 10.1145/3372297.3417872.

[KGS11] Hiltrud Kastenholz, Max Geraedts, and Hans-Konrad Selbmann. *Benchmarking im Gesundheitswesen: Ein Instrument zur Qualitätsverbesserung setzt sich durch*. In: *Zeitschrift für Evidenz, Fortbildung und Qualität im Gesundheitswesen* 105.5 (2011). DOI: 10.1016/j.zefq.2011.05.001.

[Kif09] Daniel Kifer. *Attacks on Privacy and DeFinetti's Theorem*. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. 2009. DOI: 10.1145/1559845.1559861.

[KL10] Daniel Kifer and Bing-Rong Lin. *Towards an Axiomatization of Statistical Privacy and Utility*. In: *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. 2010. DOI: 10.1145/1807085.1807106.

[KL18] Nitin Kohli and Paul Laskowski. *Epsilon Voting: Mechanism Design for Parameter Selection in Differential Privacy*. In: *IEEE Symposium on Privacy-Aware Computing (PAC)*. 2018. DOI: 10.1109/PAC.2018.00009.

[Klo+22] Susanne G. R. Klotz, Gesche Ketels, Christian A. Behrendt, Hans-Helmut König, Sebastian Kohlmann, Bernd Löwe, Johannes Petersen, Sina Stock, Eik Vettorazzi, Antonia Zapf, Inke Zastrow, Christian Zöllner, Hermann Reichenspurner, and Evaldas Girdauskas. *Interdisciplinary and cross-sectoral perioperative care model in cardiac surgery: implementation in the setting of minimally invasive heart valve surgery (INCREASE)–study protocol for a randomized controlled trial*. In: *Trials* 23.1 (2022). DOI: 10.1186/s13063-022-06455-x.

[KM04] Kaoru Kurosawa and Toshihiko Matsuo. *How to Remove MAC from DHIES*. In: *Information Security and Privacy – ACISP 2004*. 2004. DOI: 10.1007/978-3-540-27800-9_21.

[KM11] Daniel Kifer and Ashwin Machanavajjhala. *No Free Lunch in Data Privacy*. In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. 2011. DOI: 10.1145/1989323.1989345.

[KM12] Daniel Kifer and Ashwin Machanavajjhala. *A Rigorous and Customizable Framework for Privacy*. In: *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2012. DOI: 10.1145/2213556.2213571.

[Kob87] Neal Koblitz. *Elliptic curve cryptosystems*. In: *Mathematics of Computation* 48.177 (1987). DOI: https://doi.org/10.1090/S0025-5718-1987-0866109-5.

[Koh+14] Florian Kohlmayer, Fabian Prasser, Claudia Eckert, and Klaus A. Kuhn. *A flexible approach to distributed data anonymization*. In: *Journal of Biomedical Informatics* 50 (2014). DOI: 10.1016/j.jbi.2013.12.002.

[Koh96] Ron Kohavi. *Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid*. In: *International Conference on Knowledge Discovery and Data Mining (KDD'96)*. 1996. URL: https://dl.acm.org/doi/abs/10.5555/3001460.3001502.

[Kra20] Patrick Krause. *Entwicklung einer mobilen Anwendung für die Erfassung von medizinschen Daten*. Master's thesis. Universität Hamburg, 2020.

[Kra23] Maximilian Krass. *Differential Privacy für medizinische Forschungsdatenbanken*. Master's thesis. Universität Hamburg, 2023.

[Kre+21] Thea Kreutzburg, Frederik Peters, Jenny Kuchenbecker, Ursula Marschall, Regent Lee, Levente Kriston, E. Sebastian Debus, and Christian-Alexander Behrendt. *Editor's Choice – The GermanVasc Score: A Pragmatic Risk Score Predicts Five Year Amputation Free Survival in Patients with Peripheral Arterial Occlusive Disease*. In: *European Journal of Vascular and Endovascular Surgery* 61.2 (2021). DOI: https://doi.org/10.1016/j.ejvs.2020.11.013.

[KS14a] Shiva P. Kasiviswanathan and Adam Smith. *On the 'Semantics' of Differential Privacy: A Bayesian Formulation*. In: *Journal of Privacy and Confidentiality* 6.1 (2014). DOI: 10.29012/jpc.v6i1.634.

[KS14b] Martin Kroll and Simone Steinmetzer. *Automated Cryptanalysis of Bloom Filter Encryptions of Health Records*. 2014. arXiv: 1410.6739 [cs.CR].

[Kuz+11] Mehmet Kuzu, Murat Kantarcioglu, Elizabeth Durham, and Bradley Malin. *A Constraint Satisfaction Cryptanalysis of Bloom Filters in Private Record Linkage*. In: *Privacy Enhancing Technologies*. 2011. DOI: 10.1007/978-3-642-22263-4_13.

[Kuz+12] Mehmet Kuzu, Murat Kantarcioglu, Elizabeth Ashley Durham, Csaba Toth, and Bradley Malin. *A practical approach to achieve private medical record linkage in light of public resources*. In: *Journal of the American Medical Informatics Association* 20.2 (2012). DOI: 10.1136/amiajnl-2012-000917.

[LBÜ15] Martin Lablans, Andreas Borg, and Frank Ückert. *A RESTful interface to pseudonymization services in modern web applications*. In: *BMC medical informatics and decision making* 15.1 (2015). DOI: 10.1186/s12911-014-0123-5.

[LC11] Jaewoo Lee and Chris Clifton. *How Much Is Enough? Choosing ϵ for Differential Privacy*. In: *Information Security (ISC 2011)*. 2011. DOI: 10.1007/978-3-642-24861-0_22.

[LDR05] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. *Incognito: Efficient Full-Domain K-Anonymity*. In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. 2005. DOI: 10.1145/1066157.1066164.

[LDR06] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. *Mondrian Multidimensional K-Anonymity*. In: *22nd International Conference on Data Engineering (ICDE'06)*. 2006. DOI: 10.1109/ICDE.2006.101.

[Leh19] Anja Lehmann. *ScrambleDB: Oblivious (Chameleon) Pseudonymization-as-a-Service.* In: *Proceedings on Privacy Enhancing Technologies* 2019.3 (2019). DOI: 10.2478/popets-2019-0048.

[Lev66] Vladimir I Levenshtein. *Binary codes capable of correcting deletions, insertions, and reversals*. In: *Soviet Physics Doklady* 10.8 (1966). Translated from russian. Original article appeared in Doklady Akademii Nauk SSSR 163.4 (1965).

[Li+10] Jin Li, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou. *Fuzzy Keyword Search over Encrypted Data in Cloud Computing*. In: *2010 Proceedings IEEE INFOCOM*. 2010. DOI: 10.1109/INFCOM.2010.5462196.

[Li+16] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. *Differential Privacy: From Theory to Practice*. Springer, 2016. DOI: 10.1007/978-3-031-02350-7.

[Li+20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. *Federated Learning: Challenges, Methods, and Future Directions*. In: *IEEE Signal Processing Magazine* 37.3 (2020). DOI: 10.1109/MSP.2020.2975749.

[Liu+21] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. *When Machine Learning Meets Privacy: A Survey and Outlook*. In: *ACM Computing Surveys* 54.2 (2021). DOI: 10.1145/3436755.

[LL09] Tiancheng Li and Ninghui Li. *On the Tradeoff between Privacy and Utility in Data Publishing*. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2009. DOI: 10.1145/1557019.1557079.

[LLP19] Bryan Cave Leighton Paisner LLP. *California Consumer Privacy Act (CCPA) FAQs: What is 'pseudonymized' data?* 2019. URL: https://ccpa-info.com/what-is-pseudonymized-data/ (visited on 7.6.2023).

[LLV07] N. Li, T. Li, and S. Venkatasubramanian. *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*. In: *IEEE 23rd International Conference on Data Engineering*. 2007. DOI: 10.1109/ICDE.2007.3678560.

[LLV10] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. *Closeness: A New Privacy Measure for Data Publishing*. In: *IEEE Transactions on Knowledge and Data Engineering* 22.7 (2010). DOI: 10.1109/TKDE.2009.139.

[LS08] Grigorios Loukides and Jianhua Shao. *Data Utility and Privacy Protection Trade-off in k-Anonymisation*. In: *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society*. 2008. DOI: 10.1145/1379287.1379296.

[LTX08] Jiexing Li, Yufei Tao, and Xiaokui Xiao. *Preservation of Proximity Privacy in Publishing Numerical Sensitive Data*. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. 2008. DOI: 10.1145/1376616.1376666.

[LVW21] Terrance Liu, Giuseppe Vietri, and Steven Z. Wu. *Iterative Methods for Private Synthetic Data: Unifying Framework and New Methods*. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/0678c572b0d5597d2d4a6b5bd135754c-Paper.pdf.

[Mac+06] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. *l-Diversity: Privacy Beyond k-Anonymity*. In: *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*. 2006. DOI: 10.1109/ICDE.2006.1.

[Mac+08] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. *Privacy: Theory meets Practice on the Map*. In: *IEEE 24th International Conference on Data Engineering*. 2008. DOI: 10.1109/ICDE.2008.4497436.

[Mac08] Ashwin Machanavajjhala. *Defining and Enforcing Privacy in Data Sharing*. PhD thesis. Cornell University, USA, 2008. URL: https://hdl.handle.net/1813/11192.

[Mal+04] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. *Fairplay – A Secure Two-Party Computation System*. In: *13th USENIX Security Symposium (USENIX Security'04)*. 2004. URL: https://www.usenix.org/legacy/publications/library/proceedings/sec04/tech/malkhi.html.

[Mar+07] David J. Martin, Daniel Kifer, Ashwin Machanavajjhala, Johannes Gehrke, and Joseph Y. Halpern. *Worst-Case Background Knowledge for Privacy-Preserving Data Publishing*. In: *IEEE 23rd International Conference on Data Engineering*. 2007. DOI: 10.1109/ICDE.2007.367858.

[Mar+18] Matthias Marx, Ephraim Zimmer, Tobias Mueller, Maximilian Blochberger, and Hannes Federrath. *Hashing of personally identifiable information is not sufficient*. In: *SICHERHEIT 2018*. 2018. DOI: 10.18420/sicherheit2018_04.

[Mar+19] Sai Krishna Deepak Maram, Fan Zhang, Lun Wang, Andrew Low, Yupeng Zhang, Ari Juels, and Dawn Song. *CHURP: Dynamic-Committee Proactive Secret Sharing*. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*. 2019. DOI: 10.1145/3319535.3363203.

[Mar+99] Keith M. Martin, Josef Pieprzyk, Rei Safavi-Naini, and Huaxiong Wang. *Changing Thresholds in the Absence of Secure Channels*. In: *Information Security and Privacy (ACISP 1999)*. 1999. DOI: 10.1007/3-540-48970-3_15.

[McC07] Karen McCullagh. *Data Sensitivity: Proposals for Resolving the Conundrum*. In: *Journal of International Commercial Law and Technology* 2.4 (2007). URL: https://heinonline.org/HOL/P?h=hein.journals/jcolate2&i=190.

[McM+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. *Communication-Efficient Learning of Deep Networks from Decentralized Data*. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Vol. 54. 2017. URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

[McS09] Frank D. McSherry. *Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis*. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. 2009. DOI: 10.1145/1559845.1559850.

[Med24] Stanford University School of Medicine. *Collaborative Health Outcomes Information Registry (CHOIR)*. 2024. URL: https://choir.stanford.edu/ (visited on 24.1.2024).

[Men09] Alfred Menezes. *An introduction to pairing-based cryptography*. In: *Recent trends in cryptography* 477 (2009). DOI: 10.1090/conm/477/09303.

[MH11] Gregory J Matthews and Ofer Harel. *Data confidentiality: A review of methods for statistical disclosure limitation and methods for assessing privacy*. In: *Statistics Surveys* 5 (2011). DOI: 10.1214/11-SS074.

[MH22] Abdul Majeed and Seong Oun Hwang. *Rectification of Syntactic and Semantic Privacy Mechanisms*. In: *IEEE Security & Privacy* (2022). Early Access. DOI: 10.1109/MSEC.2022.3188365.

[Mir12] Ilya Mironov. *On Significance of the Least Significant Bits for Differential Privacy*. In: *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. 2012. DOI: 10.1145/2382196.2382264.

[Mir17] Ilya Mironov. *Rényi Differential Privacy*. In: *IEEE 30th Computer Security Foundations Symposium (CSF)*. 2017. DOI: 10.1109/CSF.2017.11.

[Moh+10] Noman Mohammed, Benjamin CM Fung, Patrick CK Hung, and Cheuk-Kwong Lee. *Centralized and distributed anonymization for high-dimensional healthcare data*. In: *ACM Transactions on Knowledge Discovery from Data* 4.4 (2010). DOI: 10.1145/1857947.1857950.

[Mon+15] Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. *Unique in the shopping mall: On the reidentifiability of credit card metadata*. In: *Science* 347.6221 (2015). DOI: 10.1126/science.1256297.

[Mou+18] Miranda Mourby, Elaine Mackey, Mark Elliot, Heather Gowans, Susan E. Wallace, Jessica Bell, Hannah Smith, Stergios Aidinlis, and Jane Kaye. *Are 'pseudonymised' data always personal data? Implications of the GDPR for administrative data research in the UK*. In: *Computer Law & Security Review* 34.2 (2018). DOI: https://doi.org/10.1016/j.clsr.2018.01.002.

[MP15] Yves-Alexandre de Montjoye and Alex "Sandy" Pentland. *Assessing data intrusion threats – Response*. In: *Science* 348.6231 (2015). DOI: 10.1126/science.348.6231.195-a.

[MRa+11] David M'Raihi, Johan Rydell, Mingliang Pei, and Salah Machani. *TOTP: Time-Based One-Time Password Algorithm*. RFC 6238. 2011. DOI: 10.17487/RFC6238.

[MS17] Steven Myers and Adam Shull. *Efficient Hybrid Proxy Re-Encryption for Practical Revocation and Key Rotation*. Cryptology ePrint Archive, Paper 2017/833. 2017. URL: https://eprint.iacr.org/2017/833.

[MSW99] Keith M Martin, Rei Safavi-Naini, and Huaxiong Wang. *Bounds and Techniques for Efficient Redistribution of Secret Shares to New Access Structures*. In: *The Computer Journal* 42.8 (1999). DOI: 10.1093/comjnl/42.8.638.

[MT07] Frank McSherry and Kunal Talwar. *Mechanism Design via Differential Privacy*. In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. 2007. DOI: 10.1109/FOCS.2007.66.

[MV16] Jack Murtagh and Salil Vadhan. *The Complexity of Computing the Optimal Composition of Differential Privacy*. In: *Theory of Cryptography (TCC 2016)*. 2016. DOI: 10.1007/978-3-662-49096-9_7.

[MV17] Ricardo Mendes and João P. Vilela. *Privacy-Preserving Data Mining: Methods, Metrics, and Applications*. In: *IEEE Access* 5 (2017). DOI: 10.1109/ACCESS.2017.2706947.

[MVT21] Luise Mehner, Saskia Nuñez von Voigt, and Florian Tschorsch. *Towards Explaining Epsilon: A Worst-Case Study of Differential Privacy Risks*. In: *IEEE European Symposium on Security and Privacy Workshops (Euro S&PW)*. 2021. DOI: 10.1109/EuroSPW54576.2021.00041.

[MW04] Adam Meyerson and Ryan Williams. *On the Complexity of Optimal K-Anonymity*. In: *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. 2004. DOI: 10.1145/1055558.1055591.

[NAC07] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. *Hiding the Presence of Individuals from Shared Databases*. In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. 2007. DOI: 10.1145/1247480.1247554.

[Nan+22] Priyanka Nanayakkara, Johes Bater, Xi He, Jessica Hullman, and Jennie Rogers. *Visualizing Privacy-Utility Trade-Offs in Differentially Private Data Releases*. 2022. arXiv: 2201.05964 [cs.CR].

[ND15] Maurizio Naldi and Giuseppe D'Acquisto. *Differential Privacy: An Estimation Theory-Based Method for Choosing Epsilon*. 2015. arXiv: 1510.00917 [cs.CR].

[Nea+19] Joseph P. Near, David Darais, Chike Abuah, Tim Stevens, Pranav Gaddamadugu, Lun Wang, Neel Somani, Mu Zhang, Nikhil Sharma, Alex Shan, and Dawn Song. *Duet: An Expressive Higher-Order Language and Linear Type System for Statically Enforcing Differential Privacy*. In: *Proceedings of the ACM on Programming Languages* 3.OOPSLA (2019). DOI: 10.1145/3360598.

[NF14] Arvind Narayanan and Edward W. Felten. *No silver bullet: De-identification still doesn't work*. 2014. URL: http://www.randomwalker.info/publications/no-silver-bullet-de-identification.pdf (visited on 24.5.2023).

[NH11] Thomas Neubauer and Johannes Heurix. *A methodology for the pseudonymization of medical data*. In: *International Journal of Medical Informatics* 80.3 (2011). DOI: https://doi.org/10.1016/j.ijmedinf.2010.10.016.

[NHF15] Arvind Narayanan, Joanna Huey, and Edward W. Felten. *A Precautionary Approach to Big Data Privacy*. 2015. URL: https://www.cs.princeton.edu/~arvindn/publications/precautionary.pdf (visited on 24.5.2023).

[Nie+14] Frank Niedermeyer, Simone Steinmetzer, Martin Kroll, and Rainer Schnell. *Cryptanalysis of Basic Bloom Filters Used for Privacy Preserving Record Linkage*. In: *Journal of Privacy and Confidentiality* 6.2 (2014). DOI: 10.29012/jpc.v6i2.640.

[Nie+21] Anna Niemeyer, Sebastian C. Semler, Christof Veit, Wolfgang Hoffmann, den Neeltje van Berg, Rainer Röhrig, Carolin Gurisch, Irene Schlünder, and Irina Beckedorf. *Gutachten zur Weiterentwicklung medizinischer Register zur Verbesserung der Dateneinspeisung und -anschlussfähigkeit*. 2021. URL: https://www.bundesgesundheitsministerium.de/fileadmin/Dateien/5_Publikationen/Gesundheit/Berichte/REG-GUT-2021_Registergutachten_BQS-TMF-Gutachtenteam_2021-10-29.pdf (visited on 15.4.2021).

[Nis+17] Kobbi Nissim, Aaron Bembenek, Alexandra Wood, Mark Bun, Marco Gaboardi, Urs Gasser, David R. O'Brien, Thomas Steinke, and Salil Vadhan. *Bridging the Gap between Computer Science and Legal Approaches to Privacy*. In: *Harvard Journal of Law & Technology* 31.2 (2017). URL: https://heinonline.org/HOL/P?h=hein.journals/hjlt31%5C&i=705.

[nis15] NIST. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Federal Information Processing Standards Publication 202. 2015. DOI: 10.6028/NIST.FIPS.202.

[Nor23] NordPass. *Top 200 Most Common Passwords*. 2023. URL: https://nordpass.com/most-common-passwords-list/ (visited on 27.4.2024).

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. *Smooth Sensitivity and Sampling in Private Data Analysis*. In: *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*. 2007. DOI: 10.1145/1250790.1250803.

[NS08] Arvind Narayanan and Vitaly Shmatikov. *Robust De-anonymization of Large Sparse Datasets*. In: *IEEE Symposium on Security and Privacy (SP)*. 2008. DOI: 10.1109/SP.2008.33.

[NS09] Arvind Narayanan and Vitaly Shmatikov. *De-anonymizing Social Networks*. In: *IEEE Symposium on Security and Privacy (SP*. 2009. DOI: 10.1109/SP.2009.22.

[NS10] Arvind Narayanan and Vitaly Shmatikov. *Myths and fallacies of "personally identifiable information"*. In: *Communications of the ACM* 53.6 (2010). DOI: 10.1145/1743546.1743558.

[NS19] Arvind Narayanan and Vitaly Shmatikov. *Robust de-anonymization of large sparse datasets: a decade later*. 2019. URL: https://www.cs.princeton.edu/~arvindn/publications/de-anonymization-retrospective.pdf (visited on 20.5.2023).

[NSH19] Milad Nasr, Reza Shokri, and Amir Houmansadr. *Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning*. In: *IEEE Symposium on Security and Privacy (SP)*. 2019. DOI: 10.1109/SP.2019.00065.

[NSR11] Arvind Narayanan, Elaine Shi, and Benjamin I. P. Rubinstein. *Link prediction by de-anonymization: How We Won the Kaggle Social Network Challenge*. In: *The 2011 International Joint Conference on Neural Networks*. 2011. DOI: 10.1109/IJCNN.2011.6033446.

[OA22] Ahmed El Ouadrhiri and Ahmed Abdelhadi. *Differential Privacy for Deep and Federated Learning: A Survey*. In: *IEEE Access* 10 (2022). DOI: 10.1109/ACCESS.2022.3151670.

[Och+01] Salvador Ochoa, Jamie Rasmussen, Christine Robson, and Michael Salib. *Reidentification of individuals in Chicago's homicide database: A technical and legal study*. Tech. rep. 2001.

[Ohm09] Paul Ohm. *Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization*. In: *UCLA Law Review* 57.6 (2009). URL: https://heinonline.org/HOL/P?h=hein.journals/uclalr57&i=1713.

[Ola+22] Iyiola E. Olatunji, Jens Rauch, Matthias Katzensteiner, and Megha Khosla. *A Review of Anonymization for Healthcare Data*. In: *Big Data* (2022). DOI: 10.1089/big.2021.0169.

[OMK21] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. *A Survey of the Usages of Deep Learning for Natural Language Processing*. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.2 (2021). DOI: 10.1109/TNNLS.2020.2979670.

[Par14] EU Article 29 Data Protection Working Party. *Opinion 05/2014 on Anonymisation Techniques*. 2014. URL: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf (visited on 1.11.2022).

[Ped91] Torben Pedersen. *A Threshold Cryptosystem without a Trusted Party*. In: *Advances in Cryptology — EUROCRYPT '91*. Vol. 547. 1991. DOI: doi.org/10.1007/3-540-46416-6_47.

[Pet+19] Tom Petersen, Maximilian Blochberger, Tobias Mueller, Hannes Federrath, and Christian-Alexander Behrendt. *Sichere und datenschutzgerechte Umsetzung medizinischer Register*. In: *Datenschutz und Datensicherheit - DuD* 43.8 (2019). DOI: 10.1007/s11623-019-1153-z.

[Pet18] Tom Petersen. *Datenschutzfreundliche Speicherung unternehmensinterner Überwachungsdaten mittels Pseudonymisierung und kryptographischer Schwellwertschemata*. Master's thesis. Universität Hamburg, 2018. URL: https://edoc.sub.uni-hamburg.de/informatik/frontdoor.php?source_opus=239&la=de (visited on 23.4.2024).

[Pet20] Tom Petersen. *Datenschutzgerechte und mehrseitig sichere IT-Plattformen für die medizinische Forschung*. In: *SICHERHEIT 2020*. Gesellschaft für Informatik e.V., 2020. DOI: 10.18420/sicherheit2020_13.

[PH10] Andreas Pfitzmann and Marit Hansen. *A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management*. 2010. URL: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf (visited on 2.2.2023).

[Pho+16] Le Trieu Phong, Lihua Wang, Yoshinori Aono, Manh Ha Nguyen, and Xavier Boyen. *Proxy Re-Encryption Schemes with Key Privacy from LWE*. Cryptology ePrint Archive, Paper 2016/327. 2016. URL: https://eprint.iacr.org/2016/327.

[PMB14] Andrew Paverd, Andrew Martin, and Ian Brown. *Modelling and automatically analysing privacy properties for honest-but-curious adversaries*. Tech. rep. 2014. URL: https://ajpaverd.org/publications/casper-privacy-report.pdf (visited on 25.7.2024).

[Poh+17] Geong Sen Poh, Ji-Jian Chin, Wei-Chuen Yau, Kim-Kwang Raymond Choo, and Moesfa Soeheila Mohamad. *Searchable symmetric encryption: designs and challenges*. In: *ACM Computing Surveys* 50.3 (2017). DOI: doi.org/10.1145/3064005.

[Pra21] Fabian Prasser. *Pseudonymisation in medical research: from theory to practice*. Presentation given at the IPEN webinar "Pseudonymous data: processing personal data while mitigating risks". 2021. URL: https://www.edps.europa.eu/system/files/2021-12/05_fabian_prasser_en.pdf (visited on 26.4.2024).

[PSH17] Haoyue Ping, Julia Stoyanovich, and Bill Howe. *DataSynthesizer: Privacy-Preserving Synthetic Datasets*. In: *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. 2017. DOI: 10.1145/3085504.3091117.

[PTF16] Jules Polonetsky, Omer Tene, and Kelsey Finch. *Shades of gray: Seeing the full spectrum of practical data de-intentification*. In: *Santa Clara Law Review* 56 (2016). URL: https://heinonline.org/HOL/P?h=hein.journals/saclr56&i=627.

[Puj+21] David Pujol, Yikai Wu, Brandon Fain, and Ashwin Machanavajjhala. *Budget sharing for multi-analyst differential privacy*. In: *Proceedings of the VLDB Endowment* 14.10 (2021). DOI: 10.14778/3467861.3467870.

[Qin+16] Z. Qin, H. Xiong, S. Wu, and J. Batamuliza. *A Survey of Proxy Re-Encryption for Secure Data Sharing in Cloud Computing*. In: *IEEE Transactions on Services Computing* (2016). DOI: 10.1109/TSC.2016.2551238.

[QYL14] Wahbeh Qardaji, Weining Yang, and Ninghui Li. *PriView: Practical Differentially Private Release of Marginal Contingency Tables*. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. 2014. DOI: 10.1145/2588555.2588575.

[RAP16] Slobodan Ribaric, Aladdin Ariyaeeinia, and Nikola Pavesic. *De-identification for privacy protection in multimedia content: A survey*. In: *Signal Processing: Image Communication* 47 (2016). DOI: 10.1016/j.image.2016.05.020.

[Res18] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. 2018. DOI: 10.17487/RFC8446.

[RH16] Ira S. Rubinstein and Woodrow Hartzog. *Anonymization and Risk*. In: *Washington Law Review* 91.2 (2016). URL: https://heinonline.org/HOL/P?h=hein.journals/washlr91&i=719.

[RHM19] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre de Montjoye. *Estimating the success of re-identifications in incomplete datasets using generative models*. In: *Nature Communications* 10.1 (2019). DOI: 10.1038/s41467-019-10933-3.

[Rie+20] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. *The future of digital health with federated learning*. In: *npj Digital Medicine* 3.1 (2020). DOI: 10.1038/s41746-020-00323-1.

[Roh+21] Florens Rohde, Martin Franke, Ziad Sehili, Martin Lablans, and Erhard Rahm. *Optimization of the Mainzelliste software for fast privacy-preserving record linkage*. In: *Journal of translational medicine* 19.1 (2021). DOI: 10.1186/s12967-020-02678-1.

[RP18] John M.M. Rumbold and Barbara K. Pierscionek. *What Are Data? A Categorization of the Data Sensitivity Spectrum*. In: *Big Data Research* 12 (2018). Big Data Centric Computational Intelligence in Bioinformatics and Healthcare. DOI: https://doi.org/10.1016/j.bdr.2017.11.001.

[RR14] Wullianallur Raghupathi and Viju Raghupathi. *Big data analytics in healthcare: promise and potential*. In: *Health Information Science and Systems* 2.1 (2014). DOI: 10.1186/2047-2501-2-3.

[RS00] Alexander Roßnagel and Philip Scholz. *Datenschutz durch Anonymität und Pseudonymität–Rechtsfolgen der Verwendung anonymer und pseudonymer Daten*. In: *Multimedia und Recht* 12 (2000). DOI: https://beck-online.beck.de/Bcid/Y-300-Z-MMR-B-2000-S-721-N-1.

[RSH07] Vibhor Rastogi, Dan Suciu, and Sungho Hong. *The Boundary between Privacy and Utility in Data Publishing*. In: *Proceedings of the 33rd International Conference on Very Large Data Bases*. 2007. URL: https://vldb.org/conf/2007/papers/research/p531-rastogi.pdf.

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. *The earth mover's distance as a metric for image retrieval*. In: *International journal of computer vision* 40.2 (2000). DOI: 10.1023/A:1026543900054.

[Rub93] Donald B Rubin. *Statistical disclosure limitation*. In: *Journal of official Statistics* 9.2 (1993). URL: https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/discussion-statistical-disclosure-limitation2.pdf.

[Sam01] Pierangela Samarati. *Protecting Respondents Identities in Microdata Release*. In: *IEEE Transactions on Knowledge and Data Engineering* 13.6 (2001). DOI: 10.1109/69.971193.

[SBR09] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. *Privacy-preserving record linkage using Bloom filters*. In: *BMC medical informatics and decision making* 9 (2009). DOI: 10.1186/1472-6947-9-41.

[Sch07] Bruce Schneier. *Applied cryptography: protocols, algorithms, and source code in C*. Wiley, 2007.

[Sha79] Adi Shamir. *How to share a secret*. In: *Communications of the ACM* 22.11 (1979). DOI: 10.1145/359168.359176.

[Sho+17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. *Membership Inference Attacks Against Machine Learning Models*. In: *IEEE Symposium on Security and Privacy (SP)*. 2017. DOI: 10.1109/SP.2017.41.

[Sho00] Victor Shoup. *Practical Threshold Signatures*. In: *Proceedings of the 19th International Conference on Theory and Application of Cryptographic Techniques*. 2000.

[SLL10] David Schultz, Barbara Liskov, and Moses Liskov. *MPSS: Mobile Proactive Secret Sharing*. In: *ACM Transactions on Information and System Security* 13.4 (2010). DOI: 10.1145/1880022.1880028.

[Sol06] Daniel J. Solove. *A Taxonomy of Privacy*. In: *University of Pennsylvania Law Review* 154.3 (2006). URL: http://www.jstor.org/stable/40041279.

[SOT22] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. *Synthetic Data – Anonymisation Groundhog Day*. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/stadler.

[SRP13] Lalitha Sankar, S. Raj Rajagopalan, and H. Vincent Poor. *Utility-Privacy Tradeoffs in Databases: An Information-Theoretic Approach*. In: *IEEE Transactions on Information Forensics and Security* 8.6 (2013). DOI: 10.1109/TIFS.2013.2253320.

[SS11] Paul M. Schwartz and Daniel J. Solove. *The PII Problem: Privacy and a New Concept of Personally Identifiable Information*. In: *New York University Law Review* 86.6 (2011). URL: https://heinonline.org/HOL/P?h=hein.journals/nylr86&i=1826.

[SS98a] Pierangela Samarati and Latanya Sweeney. *Generalizing Data to Provide Anonymity when Disclosing Information*. In: *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. 1998. DOI: 10.1145/275487.275508.

[SS98b] Pierangela Samarati and Latanya Sweeney. *Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression*. 1998. URL: https://dataprivacylab.org/dataprivacy/projects/kanonymity/paper3.pdf (visited on 22.5.2024).

[Sta+22] Sebastian Stammler, Tobias Kussel, Phillipp Schoppmann, Florian Stampe, Galina Tremper, Stefan Katzenbeisser, Kay Hamacher, and Martin Lablans. *Mainzelliste SecureEpiLinker (MainSEL): privacy-preserving record linkage using secure multi-party computation*. In: *Bioinformatics* 38.6 (2022). DOI: 10.1093/bioinformatics/btaa764.

[Sta17] International Organization for Standardization. *Health informatics — Pseudonymization*. Standard ISO/TS 25237:2017. 2017. URL: https://www.iso.org/standard/63553.html.

[Sta18] International Organization for Standardization. *Privacy enhancing data de-identification terminology and classification of techniques*. Standard ISO/IEC 20889:2018. 2018. URL: https://www.iso.org/standard/69373.html.

[Swe00] Latanya Sweeney. *Simple Demographics Often Identify People Uniquely*. 2000. URL: https://dataprivacylab.org/projects/identifiability/paper1.pdf (visited on 22.5.2024).

[Swe01] Latanya Sweeney. *Computational disclosure control: A primer on data privacy protection*. PhD thesis. Massachusetts Institute of Technology, 2001. URL: http://hdl.handle.net/1721.1/8589.

[Swe02] Latanya Sweeney. *k-anonymity: A model for protecting privacy*. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.5 (2002). DOI: 10.1142/S0218488502001648.

[Swe15] Latanya Sweeney. *Only You, Your Doctor, and Many Others May Know*. In: *Technology Science* (2015). URL: https://techscience.org/a/2015092903/.

[Swe97] Latanya Sweeney. *Weaving Technology and Policy Together to Maintain Confidentiality Symposium Article*. In: *Journal of Law, Medicine and Ethics* 25.2&3 (1997). URL: https://heinonline.org/HOL/P?h=hein.journals/medeth25&i=108.

[Swe98] Latanya Sweeney. *Datafly: A system for providing anonymity in medical data*. In: *Database Security XI: Status and Prospects* (1998). DOI: 10.1007/978-0-387-35285-5_22.

[SWP00] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. *Practical techniques for searches on encrypted data*. In: *IEEE Symposium on Security and Privacy (SP)*. 2000. DOI: 10.1109/SECPRI.2000.848445.

[Tao+08] Yufei Tao, Xiaokui Xiao, Jiexing Li, and Donghui Zhang. *On Anti-Corruption Privacy Preserving Publication*. In: *IEEE 24th International Conference on Data Engineering*. 2008. DOI: 10.1109/ICDE.2008.4497481.

[Tea17] Apple Differential Privacy Team. *Learning with Privacy at Scale*. 2017. URL: https://machinelearning.apple.com/research/learning-with-privacy-at-scale (visited on 13.5.2023).

[Tea20] The OpenDP Team. *The OpenDP White Paper*. 2020. URL: https://projects.iq.harvard.edu/files/opendifferentialprivacy/files/opendp_white_paper_11may2020.pdf (visited on 14.11.2023).

[Tem17] Matthias Templ. *Statistical disclosure control for microdata*. Springer, 2017. DOI: 10.1007/978-3-319-50272-4.

[TG12] Tamir Tassa and Ehud Gudes. *Secure Distributed Computation of Anonymized Views of Shared Databases*. In: *ACM Transactions on Database Systems* 37.2 (2012). DOI: 10.1145/2188349.2188353.

[Tha+20] Pratiksha Thaker, Mihai Budiu, Parikshit Gopalan, Udi Wieder, and Matei Zaharia. *Overlook: Differentially Private Exploratory Visualization for Big Data*. 2020. arXiv: 2006.12018 [cs.CR].

[TKP19] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. *DP-CGAN: Differentially Private Synthetic Data and Label Generation*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2019. URL: https://openaccess.thecvf.com/content_CVPRW_2019/html/CV-COPS/Torkzadehmahani_DP-CGAN_Differentially_Private_Synthetic_Data_and_Label_Generation_CVPRW_2019_paper.html.

[TMK08] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. *Privacy-Preserving Anonymization of Set-Valued Data*. In: *Proceedings of the VLDB Endowment* 1.1 (2008). DOI: 10.14778/1453856.1453874.

[Tri+15] Andrea C. Tricco, Elise Cogo, Wanrudee Isaranuwatchai, Paul A. Khan, Geetha Sanmugalingham, Jesmin Antony, Jeffrey S. Hoch, and Sharon E. Straus. *A systematic review of cost-effectiveness analyses of complex wound interventions reveals optimal treatments for specific wound types*. In: *BMC Medicine* 13.1 (2015). DOI: 10.1186/s12916-015-0326-3.

[TV06] T.M. Truta and B. Vinay. *Privacy Protection: p-Sensitive k-Anonymity Property*. In: *22nd International Conference on Data Engineering Workshops (ICDEW'06)*. 2006. DOI: 10.1109/ICDEW.2006.116.

[Uni24] Radboud University. *PEP - Responsible Data Sharing Repository*. 2024. URL: https://pep.cs.ru.nl/index.html (visited on 24.1.2024).

[Vat+17] Dinusha Vatsalan, Ziad Sehili, Peter Christen, and Erhard Rahm. *Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges*. In: *Handbook of Big Data Technologies*. Springer International Publishing, 2017. DOI: 10.1007/978-3-319-49340-4_25.

[Ver+16] Eric Verheul, Bart Jacobs, Carlo Meijer, Mireille Hildebrandt, and Joeri de Ruiter. *Polymorphic Encryption and Pseudonymisation for Personalised Healthcare*. Cryptology ePrint Archive, Paper 2016/411. 2016. URL: https://eprint.iacr.org/2016/411.

[Ver13] Vassilios S. Verykios. *Association rule hiding methods*. In: *WIREs Data Mining and Knowledge Discovery* 3.1 (2013). DOI: https://doi.org/10.1002/widm.1082.

[Vie+22] Giuseppe Vietri, Cedric Archambeau, Sergul Aydore, William Brown, Michael Kearns, Aaron Roth, Ankit Siva, Shuai Tang, and Steven Z. Wu. *Private Synthetic Data for Multitask Learning and Marginal Queries*. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/7428310c0f97f1c6bb2ef1be99c1ec2a-Paper-Conference.pdf.

[VSK20] Kerstin N. Vokinger, Daniel J. Stekhoven, and Michael Krauthammer. *Lost in Anonymization — A Data Anonymization Reference Classification Merging Legal and Technical Considerations*. In: *The Journal of Law, Medicine & Ethics* 48.1 (2020). DOI: 10.1177/1073110520917025.

[VW21] Salil Vadhan and Tianhao Wang. *Concurrent Composition of Differential Privacy*. In: *Theory of Cryptography Conference*. 2021. DOI: 10.1007/978-3-030-90453-1_20.

[War65] Stanley L Warner. *Randomized response: A survey technique for eliminating evasive answer bias*. In: *Journal of the American Statistical Association* 60.309 (1965). DOI: 10.1080/01621459.1965.10480775.

[WD12] Leon Willenborg and Ton De Waal. *Elements of statistical disclosure control*. Springer Science & Business Media, 2012. DOI: 10.1007/978-1-4613-0121-9.

[Wei+20] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. *Federated Learning With Differential Privacy: Algorithms and Performance Analysis*. In: *IEEE Transactions on Information Forensics and Security* 15 (2020). DOI: 10.1109/TIFS.2020.2988575.

[Wet16] Jos Wetzels. *Open Sesame: The Password Hashing Competition and Argon2*. 2016. arXiv: 1602.03097 [cs.CR].

[WF06] Ke Wang and Benjamin C. M. Fung. *Anonymizing Sequential Releases*. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2006. DOI: 10.1145/1150402.1150449.

[Whe16] Daniel Lowe Wheeler. *zxcvbn: Low-Budget Password Strength Estimation*. In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/wheeler.

[Wil+19] Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. *Differentially Private SQL with Bounded User Contribution*. 2019. arXiv: 1909.01917 [cs.CR].

[Won+06] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, and Ke Wang. *(α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy Preserving Data Publishing*. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2006. DOI: 10.1145/1150402.1150499.

[Won+07] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, and Jian Pei. *Minimality Attack in Privacy Preserving Data Publishing*. In: *Proceedings of the 33rd International Conference on Very Large Data Bases*. 2007. DOI: https://vldb.org/conf/2007/papers/research/p543-wong.pdf.

[Woo+23] Alexandra Wood, Micah Altman, Kobbi Nissim, and Salil Vadhan. *Handbook on Using Administrative Data for Research and Evidence-based Policy: Designing Access with Differential Privacy*. 2023. URL: https://admindatahandbook.mit.edu/book/v1.0/diffpriv.html#diffpriv-appendix (visited on 25.11.2023).

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. *A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces*. In: *Proceedings of the 24rd International Conference on Very Large Data Bases*. 1998. URL: http://www.vldb.org/conf/1998/p194.pdf.

[Wu13] Felix T. Wu. *Defining Privacy and Utility in Data Sets*. In: *University of Colorado Law Review* 84.4 (2013). URL: https://heinonline.org/HOL/P?h=hein.journals/ucollr84&i=1183.

[WWC16] Yunling Wang, Jianfeng Wang, and Xiaofeng Chen. *Secure searchable encryption: a survey*. In: *Journal of communications and information networks* 1 (2016). DOI: 10.1007/BF03391580.

[XT06] Xiaokui Xiao and Yufei Tao. *Anatomy: Simple and Effective Privacy Preservation*. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*. 2006. URL: http://vldb.org/conf/2006/p139-xiao.pdf.

[XT07] Xiaokui Xiao and Yufei Tao. *M-Invariance: Towards Privacy Preserving Re-Publication of Dynamic Datasets*. In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. 2007. DOI: 10.1145/1247480.1247556.

[XV16] Sofia Xesfingi and Athanassios Vozikis. *Patient satisfaction with the healthcare system: Assessing the impact of socio-economic and healthcare provision factors*. In: *BMC Health Services Research* 16.1 (2016). DOI: 10.1186/s12913-016-1327-4.

[XYT10] Xiaokui Xiao, Ke Yi, and Yufei Tao. *The Hardness and Approximation Algorithms for L-Diversity*. In: *Proceedings of the 13th International Conference on Extending Database Technology*. 2010. DOI: 10.1145/1739041.1739060.

[Yak11] Jane Yakowitz. *Tragedy of the Data Commons*. In: *Harvard Journal of Law & Technology* 25.1 (2011). URL: https://heinonline.org/HOL/P?h=hein.journals/hjlt25&i=7.

[Yan+19] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. *Federated Machine Learning: Concept and Applications*. In: *ACM Transactions on Intelligent Systems and Technology* 10.2 (2019). DOI: 10.1145/3298981.

[Yan+20] Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam. *Local Differential Privacy and Its Applications: A Comprehensive Survey*. 2020. arXiv: 2008.03686 [cs.CR].

[Yao86] Andrew Chi-Chih Yao. *How to Generate and Exchange Secrets*. In: *27th Annual Symposium on Foundations of Computer Science (SFCS 1986)*. 1986. DOI: 10.1109/SFCS.1986.25.

[Yeo+18] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. *Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting*. In: *IEEE 31st Computer Security Foundations Symposium (CSF)*. 2018. DOI: 10.1109/CSF.2018.00027.

[ZC22] Ying Zhao and Jinjun Chen. *A Survey on Differential Privacy for Unstructured Data Content*. In: *ACM Computing Surveys* 54.10s (2022). DOI: 10.1145/3490237.

[Zha+07] Qing Zhang, Nick Koudas, Divesh Srivastava, and Ting Yu. *Aggregate Query Answering on Anonymized Tables*. In: *IEEE 23rd International Conference on Data Engineering*. 2007. DOI: 10.1109/ICDE.2007.367857.

[Zha+12a] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. *Functional Mechanism: Regression Analysis under Differential Privacy*. In: *Proceedings of the VLDB Endowment* 5.11 (2012). DOI: 10.14778/2350229.2350253.

[Zha+12b] Zhifang Zhang, Yeow Meng Chee, San Ling, Mulan Liu, and Huaxiong Wang. *Threshold changeable secret sharing schemes revisited*. In: *Theoretical Computer Science* 418 (2012). DOI: 10.1016/j.tcs.2011.09.027.

[Zha+17] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. *PrivBayes: Private Data Release via Bayesian Networks*. In: *ACM Transactions on Database Systems* 42.4 (2017). DOI: 10.1145/3134428.

[Zha+18] Dan Zhang, Ryan McKenna, Ios Kotsogiannis, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. *EKTELO: A Framework for Defining Differentially-Private Computations*. In: *Proceedings of the 2018 International Conference on Management of Data*. 2018. DOI: 10.1145/3183713.3196921.

[Zha+19] Hengchu Zhang, Edo Roth, Andreas Haeberlen, Benjamin C. Pierce, and Aaron Roth. *Fuzzi: A Three-Level Logic for Differential Privacy*. In: *Proceedings of the ACM on Programming Languages* 3.ICFP (2019). DOI: 10.1145/3341697.

[Zho+05] Lidong Zhou, M. A. Marsh, F.B. Schneider, and A. Redz. *Distributed Blinding for Distributed ElGamal Re-Encryption*. In: *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. 2005. DOI: 10.1109/ICDCS.2005.24.

[Zig+20] Athanasios Zigomitros, Fran Casino, Agusti Solanas, and Constantinos Patsakis. *A Survey on Privacy Properties for Data Publishing of Relational Data*. In: *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.2980235.

[Zim+20] Ephraim Zimmer, Christian Burkert, Tom Petersen, and Hannes Federrath. *PEEPLL: Privacy-Enhanced Event Pseudonymisation with Limited Linkability*. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 2020. DOI: 10.1145/3341105.3375781.

[ZK17] Danfeng Zhang and Daniel Kifer. *LightDP: Towards Automating Differential Privacy Proofs*. In: *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*. 2017. DOI: 10.1145/3009837.3009884.

[ZXL18] Rui Zhang, Rui Xue, and Ling Liu. *Searchable Encryption for Healthcare Clouds: A Survey*. In: *IEEE Transactions on Services Computing* 11.6 (2018). DOI: 10.1109/TSC.2017.2762296.

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. Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Hamburg, den 24. September 2024

Tom Petersen