SELBSTDATENSCHUTZ DURCH PRÄVENTIVE VERARBEITUNGSKONTROLLE

Dissertation

zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)

der Fakultät für Mathematik, Informatik und Naturwissenschaften,
Fachbereich Informatik
der Universität Hamburg

vorgelegt von
Diplom-Wirtschaftsinformatiker
KAI WAGNER

Hamburg, 2013

Erster Gutachter: Prof. Dr.-Ing. Hannes Federrath, Universität Hamburg
Zweiter Gutachter: Prof. Dr. Günther Pernul, Universität Regensburg

Tag der Disputation: 14. Dezember 2012

Kurzfassung

Diese Arbeit untersucht den Einsatz von Maßnahmen zur Verarbeitungskontrolle im Dienst des Datenschutzes. Dabei steht die technische Durchsetzung von zweckgebundener und anlassbezogener Datenverarbeitung im Vordergrund: Personenbezogene Daten sollen Verarbeitern nur dann verfügbar gemacht werden, wenn sie tatsächlich benötigt werden, so spät wie möglich und nur unter explizit vorliegendem Einverständnis des Betroffenen. Letzterer soll größtmögliche Kontrolle über die Regeln zur Nutzung seiner Daten erhalten.

Um datenschutzfreundliche Techniken hinsichtlich dieser Anforderung systematisch zu untersuchen, werden zunächst geeignete Bewertungskriterien definiert. Die erarbeiteten Kriterien finden sodann ihre Anwendung auf ausgewählte bestehende Ansätze. Anhand der Evolution der bisherigen Verfahren und der identifizierten Lücken zeigt sich das Potential für ein neues Verfahren mit zusätzlichen Fähigkeiten.

Mit dem „Purpose sensitive data provisioning guard“ (PDG) wird schließlich ein solches Verfahren vorgeschlagen, mit dem sich Daten sicher zwischen den Betroffenen und Verarbeitern austauschen lassen, ohne dass sich beide Parteien kennen müssen. Die Herausgabe von Daten unterliegt dabei strikten Anforderungen an den Verarbeiter: Er muss seine grundsätzliche Eignung als Empfänger nachweisen, den Zweck der Verarbeitung darlegen und die lückenlose Protokollierung seiner Datenabrufe akzeptieren. Seine Anfragen werden pro Datentyp gegen Policies geprüft, die der Betroffene erstellt hat, und die den Datenzugriff des Verarbeiters gestatten oder verweigern. Das Verfahren bedient sich dazu der Kombination etablierter kryptographischer Techniken, um einen sicheren Schlüsselaustausch zwischen einander nicht bekannten Parteien abzubilden und blinde Policy-Abgleiche durch Dritte vornehmen zu lassen.

Abstract

This thesis researches how processing controls can be leveraged for the benefit of data privacy. Its focus is to technically enforce that personal data is used only for a specified purpose and on defined events: Personal data must only be exposed to a processor if it is truly required, at the latest moment possible and only with the explicit consent of the affected person. The latter should to the highest extent be in control of the rules applying to the use of his data.

To systematically test privacy enhancing technologies for these requirements, suitable evaluation criteria are defined first. Subsequently these criteria are used to assess selected existing approaches. From the evolution of the existing systems and the identified gaps in privacy protection, the potential for a new approach with additional capabilities emerges.

The „purpose sensitive data provisioning guard“ (PDG) is the proposal for such a new approach. It enables data to be exchanged safely between owner and processor, even if both do not know each other. With this approach the exposure of personal data poses strong requirements on the processors: They must prove their general qualification, expose their intention for using the actually requested data, and accept comprehensive logging of their requests. For each data type their requests are tested against the policies created by the data owner, approving or denying the processors' access. The system uses a combination of existing cryptographic techniques to enable secure key exchange between parties not knowing each other, and it supports blind policy matching through third parties.

Danksagung

Meinem Doktorvater Prof. Dr.-Ing. Hannes Federrath gilt mein ausdrücklicher Dank für seine Bereitschaft zur Betreuung meiner externen Promotion, für die motivierende und geduldige Begleitung meiner Arbeit über die Jahre und für die Impulse, die er dem Projekt in unseren Gesprächen gegeben hat.

Ebenso danke ich herzlich meinem Co-Betreuer Prof. Dr. Günther Pernul, sowie Karl-Peter Fuchs, Christoph Gerber und Dominik Herrmann für anregende Diskussionen und wertvolle Hinweise.

Christian Baumann, Ahmed Hodjov, Stephan Hoepfner, Roman Jerger, Dennis Keitzel, Hannes Kuhlmann und Stephan Lauterbach haben das von mir entworfene Verfahren im Rahmen ihrer Masterarbeit prototypisch implementiert. Vielen Dank für die daraus resultierenden Einblicke in die praktische Nutzbarkeit meiner Protokolle und die pragmatischen Optimierungsvorschläge.

Eine berufsbegleitende Promotion kann nur mit viel Verständnis und Unterstützung im Privaten gelingen. Ich hatte das Glück, in meiner Familie, bei meinen Schwiegereltern und allen voran bei meiner Frau Katja auf beides zu treffen. Danke für diesen großartigen Rückhalt.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Individuen wollen oder müssen in ihren gesellschaftlichen Rollen, etwa als Angestellter, Patient, Kunde, Steuerzahler oder Vereinsmitglied, personenbezogene Daten für eine Vielzahl möglicher Verarbeiter bereithalten. Ein Missbrauch dieser Daten durch die Empfänger oder Dritte wird strafrechtlich verfolgt, denn die Verarbeitung darf nur auf gesetzlicher Grundlage oder mit der Einwilligung der Betroffenen erfolgen[1]. Die Datenschutzgesetze verbieten die zweckfremde Nutzung personenbezogener Daten. Wer seine Rechte diesbezüglich verletzt sieht, kann sich an Datenschutzbeauftragte oder Aufsichtsbehörden wenden. Dennoch zeigt die Zunahme alltäglicher bedenklicher Datennutzung[2], beispielsweise beim Zielgruppen-Marketing[3], ebenso wie die Berichterstattung über Fälle umfangreichen Datenmissbrauchs[4], dass der Schutz der Privatheit unzureichend ist.

Viele Organisationen sind bereit, Strafen in Kauf nehmen, wenn der Vorteil durch die Nutzung der Daten ausreichend hoch ist. Zudem ist die Beweisbarkeit eines Missbrauchs, wenn überhaupt möglich, mit hohem Aufwand für den Geschädigten verbunden. Datenschutz als juristische Maßnahme zur Sanktionierung begangener Verstöße ist erforderlich, aber nicht ausreichend. Möglichst weitgehender präventiver, technischer Datenschutz wird zusätzlich benötigt. Zum einen dürfen sensible Daten erst dann preisgegeben werden, wenn der gesetzlich oder einvernehmlich festgelegte Anlass ihrer Nutzung eintritt. Sie dürfen nur dem beabsichtigten Verarbeiter zugänglich gemacht werden, der seinen Informationsbedarf zu legitimieren hat. Zum anderen bedarf auch die Verfolgung von stattgefundenem Missbrauch technischer Unterstützung, um nicht zu leugnende Beweise für die Nutzung der Daten und die Identität des Verarbeiters zu schaffen.

Eine möglichst exakte Definition des Anlasses zur Nutzung (Zeitpunkt und Verwendungszweck), hat große Bedeutung, da heute davon auszugehen ist, dass einmal gegenüber einem Verarbeiter offen gelegte personenbezogene Daten technisch nicht mehr zuverlässig vor Missbrauch geschützt werden können, solange kein unmittelbarer Einfluss auf die Infrastruktur und verwendeten Verfahren der Verarbeiter besteht[5]. Somit sind die Mög-

[1] BDSG §4(1)

[2] Die Anzahl schriftlicher Eingaben an den Bundesbeauftragten für den Datenschutz hat sich beispielsweise von 1647 Eingaben im Jahr 2002 auf 6087 Eingaben im Jahr 2010 fast vervierfacht, vgl. [Scha11].

[3] Bruce Schneier in einem cnet-Interview 2009 dazu auf die Frage nach der größten Bedrohung für die Privatsphäre: „Marketing. The legal collection, storage, resale, and reuse of personal information. Information brokers are doing more to hurt consumer privacy than anything criminals or the government can do. And, even worse, the government can buy information from them, and criminals can break into their databases. “, http://news.cnet.com/8301-27080_3-10381460-245.html (Zugriff: 13.05.2012).

[4] Das Privacy Rights Clearinghouse führt etwa eine Liste mit mehr als 3000 Vorfällen in den USA seit 2005, bei denen über 560 Millionen personenbezogene Datensätze unrechtmäßig weitergegeben wurden, http://www.privacyrights.org/data-breach (Zugriff: 13.05.2012).

[5] Vielmehr ist nach dem Offenlegen von Daten der Grad ihrer Vertraulichkeit streng monoton fallend. Ein einmal aufgedeckter Inhalt kann also nie wieder vertraulich gemacht werden, vgl. [JaSc04].

lichkeiten des Selbstdatenschutzes nach der Offenlegung der Daten begrenzt. Bis zu diesem Punkt jedoch sind geeignete Mittel erforderlich, allgemein verbindliche wie auch individuell formulierte Datenschutzregeln wirksam durchzusetzen.

1.2 Forschungsfragen

Das Ziel der vorliegenden Arbeit ist es, Antworten auf die folgenden Forschungsfragen zu erarbeiten:

1. Welche Optionen für die technische Durchsetzung von Datenschutzregeln bestehen und wie kann man sie systematisieren?

2. Welche Aussagen kann man daraus über die Qualität entsprechender Verfahren im Sinne des Datenschutzes treffen?

3. Wie sind bestehende Verfahren der Praxis und theoretische Ansätze in die Systematik einzuordnen? Wie werden sie bewertet?

4. Welche Entwicklungsziele sollten angegangen werden, um ein höheres Datenschutzniveau zu erreichen?

5. Welche Anforderungen soll ein Verfahren erfüllen, das im Bereich der bewussten Datenbereitstellung für situativ bedingte Abrufe dieses Niveau anstrebt?

6. Wie ist ein solches Verfahren zu gestalten? Welche Protokolle, Algorithmen und Systemkomponenten eignen sich zum Einsatz?

7. Wie verhält sich das entwickelte Verfahren gegenüber Angriffen von Teilnehmern und von Außenstehenden?

8. Erreicht das neue Verfahren die angestrebte Verbesserung im Datenschutz-Niveau? Welcher weitere Entwicklungsbedarf besteht?

1.3 Vorgehen und Kapitelstruktur

Kapitel 2 untersucht zunächst, unter welchen Umständen Individuen ihre personenbezogenen Daten in der digitalen Welt Anderen zur Verfügung stellen und mit welchen Strategien dabei den Herausforderungen im Bereich des Datenschutzes begegnet wird. Im Speziellen wird dargestellt, wie Verarbeitungskontrolle als datenschutzfreundliche Technik ein wirksames Mittel zum Selbstdatenschutz sein kann. Die im weiteren Verlauf der Arbeit beschriebenen Verfahren zur Verarbeitungskontrolle bedürfen, um Vergleichbarkeit zu gewährleisten, eines gemeinsamen Bewertungsrahmens. Er wird ebenfalls in diesem Kapitel erarbeitet. Durch universelle Gestaltung leistet er auch über

den Rahmen dieser Arbeit hinaus einen Beitrag zur Systematisierung datenschutzfreundlicher Techniken.

Ausgewählte Ansätze zur Verarbeitungskontrolle werden in Kapitel 3 dargestellt. Sie werden anhand der Kriterien des zuvor definierten Bewertungsrahmens untersucht und dabei auch hinsichtlich ihrer technischen und organisatorischen Umsetzbarkeit geprüft. Es wird der Pfad skizziert, entlang dessen sich die bislang realisierten Verfahren entwickelt haben und schließlich gezeigt, welche Eigenschaften im nächsten Entwicklungsschritt hinzugefügt werden sollten, um das Datenschutzniveau zu erhöhen.

Hieraus leitet sich die Motivation eines neuen Verfahrens in Kapitel 4 ab. Es werden die konkreten Anforderungen an das Verfahren aus den zuvor beschriebenen Entwicklungen abgeleitet und durch beispielhafte Einsatzszenarien veranschaulicht.

Kapitel 5 stellt dann das neue Verfahren „Purpose sensitive data provisioning guard“ (PDG) vor. Um die verwendeten Techniken sukzessive einzuführen, wird zunächst die Grundform des Verfahrens PDG dargestellt, und anschließend zu PDG/v („PDG with variable processor groups“) vervollständigt. Das Kapitel definiert die Protokolle zur Initialisierung des Verfahrens, zum Schlüsselaustausch und zur Datenweitergabe.

In Kapitel 6 werden Möglichkeiten zur Implementierung des Verfahrens in der Grundform und im vollständigen Szenario beschrieben. Insbesondere widmet es sich der Auswahl des geeigneten Krypto-Algorithmus für das zentrale Protokoll-Element des PDG/v, den Schlüsselaustausch zwischen Betroffenem und dem vorab nicht bekannten Mitglied einer Empfängergruppe. Zudem schlägt es geeignete Optionen zur Applikationsarchitektur vor. Das Kapitel schließt mit einer zusammenfassenden Beschreibung der prototypischen Implementierung von PDG/v an der Universität Hamburg.

Kapitel 7 analysiert PDG und PDG/v hinsichtlich ihrer Widerstandsfähigkeit gegenüber Angriffen der Protokoll-Teilnehmer und durch Außenstehende. Es arbeitet die zentralen Risikofaktoren heraus, und benennt die verfügbaren Ansätze zu ihrer Linderung.

Die Arbeit schließt mit einer Zusammenfassung der erarbeiteten Erkenntnisse in Kapitel 8 anhand der eingangs formulierten Forschungsfragen. Es ordnet PDG/v in die Systematik aus Kapitel 2 ein und bewertet das Datenschutzniveau des Verfahrens.

2 Umsetzung von Datenschutzregeln

2.1 Verarbeitungskontrolle als bewusste Entscheidung

Bewegen sich Menschen in der der digitalen Welt, hinterlassen sie personenbezogene und -beziehbare[6] Spuren in den Applikationen und Netzen, in denen sie aktiv sind. Betrachtet man, auf welche Arten und mit welchen Zielen diese Daten bereitgestellt werden, lassen sich die folgenden Kategorien identifizieren:

- **Kategorie „Cookie“**

  Hierunter fallen die Datenbereitstellungen, die ein Betroffener nicht primär mit seiner Nutzung des Systems bezweckt oder die durch die Verknüpfung verschiedener von ihm bereitgestellter Daten ohne sein Wissen geschehen. Vertreter sind die Aufzeichnung und Auswertung des Surf- und Nutzungsverhaltens (sog. „Tracking“ durch Cookies[7]), Sammeln von Daten der verwendeten Hard- und Softwareausstattung, sowie die Protokollierung von Kommunikation mit anderen Nutzern[8].

  Diese Kategorie ist dadurch gekennzeichnet, dass die Datenbereitstellung für den Betroffenen weitgehend unbemerkt erfolgt und die Inhalte durch den Betreiber des Systems und nicht durch den Betroffenen definiert sind. Dementsprechend liegt der Fokus im Umgang mit dieser Art der Datenbereitstellung aus Sicht des Betroffenen auf Strategien, sie zu verhindern oder zumindest einzuschränken[9].

- **Kategorie „Facebook“**

  In Bezug auf die freiwillige Herausgabe von Daten auf der entgegengesetzten Seite der Skala steht die Art der Datenbereitstellung, wie sie durch Nutzer sozialer Netzwerke mit stetig wachsendem Engagement betrieben wird. Persönliche Daten werden in vollem Bewusstsein (sofern es sich um die eigenen Daten handelt) bereitgestellt, wobei der Zweck eher in der heterogen motivierten Formung eines Selbstbildes[10] zu suchen ist als in konkreten Nutzungszwecken. Die vorherrschende Strategie im Umgang mit der Datenbereitstellung ist daher bei den

[6] „Personenbezug: Die Daten sind eindeutig einer natürlichen Person zugeordnet, anhand der Daten ist die Person identifiziert. Personenbeziehbarkeit bezeichnet nur die prinzipielle Möglichkeit, einen Bezug zwischen den Daten und einem Betroffenen herstellen zu können. Erst unter Zuziehung weiterer Daten kann die zugeordnete Person ermittelt/identifiziert werden.“ [Sael04]

[7] vgl. [ULD11]

[8] vgl. [Roes00], [SpPo11]

[9] Allerdings wiesen schon früh Untersuchungen darauf hin, dass viele Benutzer keine spezifischen Einstellungen zu Cookies vornehmen, da diese komplex sind und der Prozess als langwierig und lästig wahrgenommen wird (vgl. [Zieg02], [EPJu00]).

[10] Karla und Gronenschild nennen unter anderem: Die Attraktivität von Neuartigem, das Streben nach Gemeinschaft und Austausch, den Hang zur Selbstoffenbarung, Flucht vor dem Alltag, Gruppendruck, wirksame Anreizsysteme und das Fehlen unmittelbar erfahrbarer Konsequenzen aus der Preisgabe personenbezogener Daten [KaGr11].

Betroffenen noch das Inkaufnehmen der Missbrauchspotentiale durch andere Nutzer des Netzwerks beziehungsweise den Betreiber[11].

- **Kategorie „Applikation“**

  Die dritte Kategorie wird im Folgenden den Betrachtungsgegenstand dieser Arbeit darstellen. Sie ist dadurch charakterisiert, dass der Betroffene die bereitzustellenden Inhalte selbst festlegt und die Bereitstellung bewusst und zur Erfüllung eines konkreten Nutzungszwecks veranlasst. Dies kann beispielsweise geschehen, weil der Betroffene aufgrund rechtlicher oder vertraglicher Verpflichtungen Daten für Andere bereit stellen muss, wie es bei der Überlassung von Daten über Familienstand, Religionszugehörigkeit, Behinderungen und anderen an die Personalabteilung des Arbeitgebers geschieht – mit dem Zweck, diesem die korrekte Berechnung des Lohnsteuerabzugs vom Gehalt zu ermöglichen. Auch freiwillige Datenbereitstellung fällt in diese Kategorie, wie sie etwa im Rahmen der Mitwirkung in einem Verein, der Teilhabe an Forschung und Bildung, der Verbesserung der eigenen Gesundheitsversorgung oder allgemeiner formuliert der zielgerichteten Ausgestaltung des persönlichen Lebensmodells erfolgt. Die Kategorie ist in der vorliegenden Arbeit von besonderem Interesse, weil die Zielgerichtetheit der Datenbereitstellung geradezu fordert, andere als den beabsichtigten Verarbeiter und andere als den gewünschten Verarbeitungszweck auszuschließen. Hier muss also das Gebot der Zweckbindung bei der Verarbeitung personenbezogener Daten vordringliche Beachtung finden. Die Natur der Daten erfordert zudem hohe Sensibilität hinsichtlich ihrer Sicherheit. Demzufolge ist möglichst weitreichende Verarbeitungskontrolle die Strategie der Wahl für Daten der Kategorie „Applikation“.

Abbildung 1 fasst die beschriebenen Kategorien mit ihren wesentlichen Eigenschaften zusammen.

Abbildung 1: Drei Kategorien der Datenbereitstellung

[11] Zudem zeigt etwa Ulbricht, dass Facebook durch entsprechende Gestaltung der Funktionen dafür sorgt, dass Benutzer ihre Datenschutzeinstellungen möglichst wenig restriktiv wählen, vgl. [Ulbr11].

Verarbeitungskontrolle ist das Regeln und Überwachen der Nutzung digitaler Inhalte.

Der Wortbestandteil „Kontrolle“ geht hier über die klassische betriebswirtschaftliche Definition hinaus, wo er sich mit der Ermittlung und Analyse von Abweichungen zwischen geplanten und realisierten Größen befasst[12], somit keinen aktiven Einfluss auf die Größen ausübt. Kontrolle in der hier verwendeten Ausprägung folgt dem angloamerikanischen „control“[13], das die Kontrollausübung, also aktive Einflussnahme, inkludiert.

Der Wortbestandteil „Verarbeitung“ bedarf ebenfalls einer Wahl aus den bestehenden Definitionen. Das Bundesdatenschutzgesetz trennt strikt zwischen verschiedenen Qualitäten des Umgehens mit Daten: Erheben, Verarbeiten und Nutzen[14]. Für alle drei Kategorien zusammen verwendet es den Begriff „Umgang“. Verarbeiten und Nutzen werden zu „Verwendung“ zusammengefasst. Nutzung ist dabei als Auffangtatbestand für die Verwendung personenbezogener Daten definiert, bei der er sich nicht um Verarbeitung – also Speichern, Verändern, Übermitteln, Sperren und Löschen – handelt[15], beispielsweise das Abgleichen oder Veröffentlichen[16].

Die EU-Datenschutzrichtlinie favorisiert den Begriff der „Verarbeitung“ als den umfassenderen, nämlich: „jeden mit oder ohne Hilfe automatisierter Verfahren ausgeführten Vorgang […] im Zusammenhang mit personenbezogenen Daten wie das Erheben, das Speichern, die Organisation, die Aufbewahrung, die Anpassung oder Veränderung, das Auslesen, das Abfragen, die Benutzung, die Weitergabe durch Übermittlung, Verbreitung oder jede andere Form der Bereitstellung, die Kombination oder die Verknüpfung sowie das Sperren, Löschen oder Vernichten.“[17], schließt also im Gegensatz zum Bundesdatenschutzgesetz ausdrücklich die Erhebung und Nutzung in die Verarbeitung ein.

Landesdatenschutzgesetze folgen einmal der Begriffswelt aus dem Bundesdatenschutzgesetz, etwa das Bayerische Datenschutzgesetz (BayDSG), andere wie das Hessische Datenschutzgesetz (HDSG) lehnen sich bei der Definition von „Verarbeitung“ an die EU-Richtlinie an.

Im Weiteren wird „Verarbeitung“ im Sinne der EU-Richtlinie verwendet.

[12] vgl. Gabler Wirtschaftslexikon, Stichwort: Kontrolle [Gabl12]
[13] „control: power or authority to direct, order or limit“ (Oxford Dictionary)
[14] BDSG §1(1) und (2)
[15] BDSG §3(4) und (5)
[16] vgl. [GoSK05]
[17] Richtlinie 95/46/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr.

2.2 Einsatz zum Selbstdatenschutz

Soll Verarbeitungskontrolle im Sinne des Datenschutzes wirtschaftlich eingesetzt werden, so sind Art, Maß und Reichweite der Kontrollmaßnahmen situativ zu wählen.

Der Schutzbedarf, den Betroffene haben, wenn es um die Verarbeitung ihrer personenbezogenen Daten geht, ist subjektiv[18]. Er verhält sich aber in jedem Fall reziprok zum Vertrauen des Betroffenen in den Verarbeiter und den Kontext der Verarbeitung[19]. Er bestimmt sich aus zwei Parametern: Dem Missbrauchspotential und der wahrgenommenen Transparenz der Datenverarbeitung. Das bedeutet, größtmögliches Vertrauen wird erreicht, wenn ein Verfahren Missbrauch der Daten mit wirkungsvollen Mitteln einschränkt und gleichzeitig dem Betroffenen die auf seine Daten bezogenen Aktivitäten anderer zeitnah und umfänglich sichtbar macht.

Ein Konzept zur Verarbeitungskontrolle muss einerseits den Schutzbedarf des Betroffenen bedienen und sich andererseits den technischen, juristischen und wirtschaftlichen Restriktionen unterwerfen. Theoretisch perfekter Datenschutz, der mit seinen Implementierungskosten den Betroffenen ruiniert, wäre ebenso wenig zielführend wie ein sicheres Verfahren, das vertragliche Rechte der Verarbeiter oder Gesetze verletzt.

Besteht nun ein Verfahren zur Verarbeitungskontrolle, bewirkt es in seiner Anwendung Rückkopplungen auf den für den Betroffenen erreichten Datenschutz und auf die von ihm wahrgenommene Transparenz der Datenverarbeitung. Im erwünschten Fall steigen Datenschutzniveau und Transparenz, was wiederum durch die resultierende Beschränkung der Missbrauchspotentiale das Vertrauen des Betroffenen in die Verarbeitung stärkt. Daraufhin kann er eine Neubewertung des subjektiven Schutzbedarfs vornehmen und das Verfahren entweder weiterführen oder Anpassungen fordern. Ohnehin bedarf die technische und gesellschaftliche Weiterentwicklung einer regelmäßigen Überprüfung der Datenschutzpräferenzen jedes Einzelnen. Auch Ereignisse im persönlichen Umfeld des Betroffenen sowie veröffentlichte Datenschutz-Vorfälle greifen in die Bewertung des Schutzbedarfs ein. Den geschilderten Regelkreis zeigt modellhaft Abbildung 2.

Der Subjektivität des Schutzbedarfs folgt zwingend, dass Datenschutz nicht alleine einer übergeordneten Instanz, dem Gesetzgeber, zur Regelung überlassen werden kann. Dessen Aufgabe ist es, allgemein verbindliche Rahmenbedingungen zu schaffen, in deren Grenzen der Einzelne entscheiden kann, welche Balance er zwischen den Vorteilen der

[18] vgl. [Schr01] zur Bedeutung des subjektiven Sicherheitsempfindens: „Almost as relevant as experts' opinions about the security of a system is the user's confidence that using the system will not endanger his privacy. "

[19] „Vertrauen ist der Glaube in den Willen und die Fähigkeit eines Anderen, sich im gegebenen Kontext und zur entsprechenden Zeit so zu verhalten, wie es der Vertrauende erwartet." Übersetzung frei nach [DiCH04].

Verwendung seiner Daten durch Andere[20] einerseits und den damit verbundenen Risiken andererseits wählt. Es bedarf also wirksamer Mittel zum Selbstdatenschutz, mit denen sich die subjektiven Entscheidungen auch über das einklagbare Recht hinaus konkretisieren und durchsetzen lassen[21]. Technisch umgesetzte Verfahren der Verarbeitungskontrolle zählen zu diesen Mitteln.

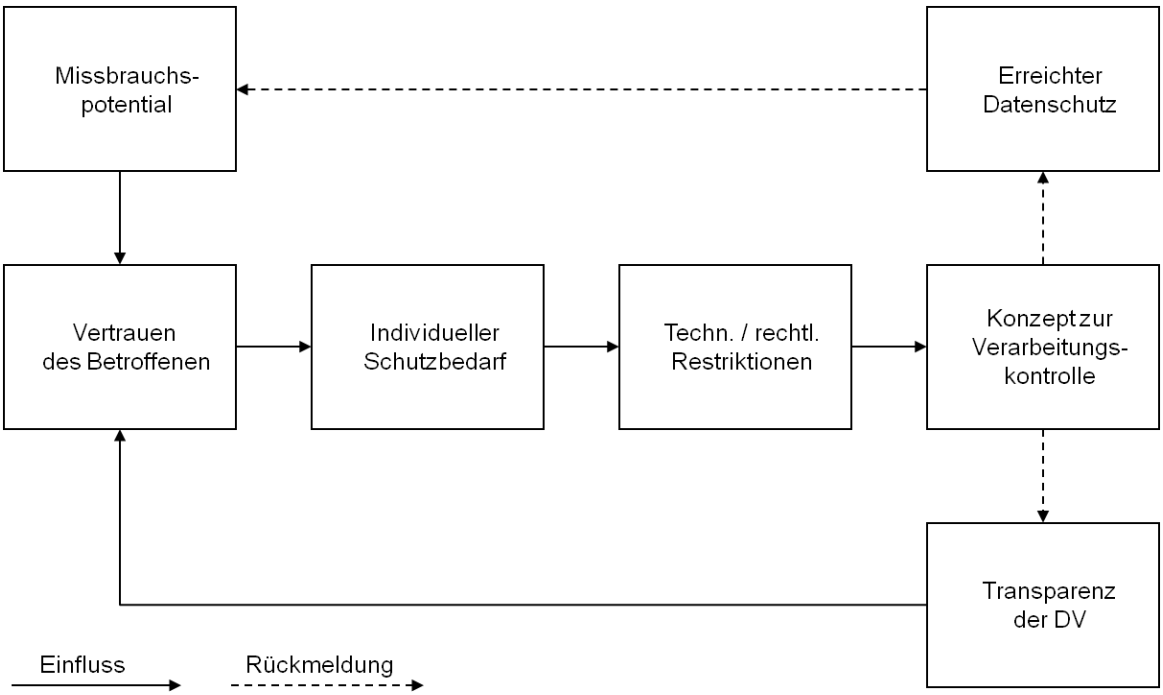


Abbildung 2: Regelkreis für Konzepte der Verarbeitungskontrolle

> *Selbstdatenschutz* ist die Menge von Aktivitäten, die ein Betroffener aktiv zum Schutz seines Rechts auf informationelle Selbstbestimmung ergreifen kann.

Datenschutzfreundliche Techniken helfen dabei, die Rechte der Betroffenen in diesem Sinne durchzusetzen.

---

[20] Posner verweist auf die mikroökonomische Betrachtungsweise, nach der jeder der Eigentümer seiner Daten sei und diese frei verkaufen könne [Posn84]. So würde mit den Mitteln des Marktes erreicht, dass Information die wert-angemessenste Verwendung findet. Das wohl plakativste Beispiel dafür, iFAY der Cocus AG, beschreibt Grimm [Grim03] (vgl. auch http://www.heise.de/newsticker/meldung/Geld-fuer-Deine-Daten-18868.html, Zugriff am 03.06.2012). Demgegenüber weist Weichert darauf hin, dass bei dem Versuch, die Nutzung personenbezogener Daten ökonomisch auszuhandeln, nicht von Chancengleichheit gesprochen werden könne. Etwaige Machtgefälle im Rahmen eines Beschäftigungsverhältnisses oder psychologisch besonders ausgeklügelte Vorgehensweisen der Datensammler stünden dem entgegen [Weic00].

[21] Bizer dazu: „Im modernen Datenschutz ergänzen sich der auf die Sicherheit und den Schutz des Gesamtsystems gerichtete *Systemdatenschutz* sowie der in der Hand des Betroffenen und Nutzers befindliche *Selbstdatenschutz*, mit dessen rechtlichen und technischen Instrumenten der Nutzer unter der Voraussetzung einer ausreichenden Information selbst sein Datenschutzniveau bestimmt.“ [Bize04]

> *Datenschutzfreundliche Techniken* (Privacy Enhancing Technologies – PET) sind Maßnahmen, die in Informationssystemen die Privatsphäre von Betroffenen schützen, indem sie die Verwendung personenbezogener Daten unnötig machen oder reduzieren beziehungsweise deren ungewollte und missbräuchliche Verarbeitung verhindern – bei gleichzeitigem Erhalt der Funktionalität des Informationssystems[22]. Als präventive Maßnahmen zur Durchsetzung von Datenschutzregeln sind datenschutzfreundliche Techniken außerdem dazu geeignet, die Aufwände nachgelagerter Kontrollen (Audits) zu reduzieren.[23]

Die Anwendung datenschutzfreundlicher Techniken beginnt im besten Fall dort, wo auch der Umgang mit den personenbezogenen Daten seinen Anfang nimmt, bei der Datenerhebung[24]. Wie oben dargestellt, ist die Erhebung juristisch bereits Schutzgegenstand der einschlägigen Datenschutzgesetze. Und sie ist ebenfalls aus technisch-organisatorischer Perspektive die erste Verteidigungslinie. Insbesondere wenn die Datenerhebung wie im Fall der geschilderten Kategorie „Applikation" direkt, also unter bewusster Mitwirkung des Betroffenen erfolgt, stellt sie dessen Eingriffsmöglichkeit dar, um das Gebot der Datensparsamkeit[25] in seinem Interesse umzusetzen. In diesem Sinne steht der Datenerhebung als Aktivität des Datenverarbeiters die Datenbereitstellung als deren Gegenpart aufseiten des Betroffenen gegenüber.

> *Datenbereitstellung* umfasst die aktive Festlegung personenbezogener Daten in Form und Inhalt, sowie das Hinzufügen von Verwaltungsinformationen durch den Betroffenen zum Zweck der Erhebung durch einen Datenverarbeiter.

[22] „Der Technik ist nicht lediglich abwehrrechtlich zu begegnen; vielmehr kann die Technik selbst zum Datenschutz beitragen." [Baer02a]

[23] Nach [BlBO03]: „PET stands for a coherent system of ICT measures that protects privacy by eliminating or reducing personal data or by preventing unnecessary and/or undesired processing of personal data, all without losing the functionality of the information system. […] Ultimately PET might replace inspections for enforcing the privacy regulations. Enforcing by means of inspections or audits to verify whether all organisations that collect personal data are complying with the privacy regulations is rather time-consuming and thus expensive. "

[24] Das bedeutet, dass bereits das Design der Erhebung diesen Grundsätzen folgt. Dazu auch [Bäum02]: „Die Grundsätze der Datenvermeidung und Datensparsamkeit setzen bereits auf der Ebene der Technikgestaltung und bei der Organisation des Verarbeitungsprozesses […], also früher als das Erforderlichkeitsprinzip an. […] Beide Grundsätze gehören deshalb in den Kreis der Überlegungen, wie der Datenschutzgedanke auch technisch effektiver umgesetzt werden könnte. […] Da Datenverarbeitungssysteme zunehmend nicht nur für eine, sondern für verschiedene Aufgaben eingesetzt werden, müssen sie unter dem Aspekt der Datenvermeidung und der Datensparsamkeit vor allem flexibel sein. Nur so kann erreicht werden, dass nicht das Verfahren mit dem umfangreichsten Datenbedarf das Niveau für das gesamte Datenverarbeitungssystem bestimmt."

[25] vgl. §3a BDSG

2.3 Dimensionen technischen Datenschutzes

Die Notwendigkeit, Daten über die reine Kontrolle des Systemzugangs hinaus zu schützen, die Sicherheit also direkt an den schützenswerten Daten zu verankern, führt zur „Data-Centric Security“[26]. Sie betrachtet in einem ganzheitlichen Ansatz den individuellen Wert von Daten für den Eigner und definiert darauf die geeigneten Verarbeitungskontrollen[27]. Paul Stamp et al. von Forrester Research konkretisieren, dass die traditionelle Infrastruktur-Sicherheit nicht obsolet, sondern die Grundvoraussetzung datenzentrischer Sicherheit ist[28]. Die Verfahren verstehen sich demnach als Ergänzungen zur nach wie vor notwendigen Sicherung des Systemperimeters.

Zur Beschreibung technischer Verarbeitungskontrolle im Bereich des Datenschutzes kann man die fünf Dimensionen verwenden, die Abbildung 3 zeigt. Diese sind:

- Gestaltung von Verarbeitungsregeln (Policies),
- Formulieren der Regeln,
- Bindung der Regeln an die zu schützenden Daten,
- Durchsetzung der Regeln im Lebenszyklus der Daten,
- Technische Implementierung der Verarbeitungskontrolle.

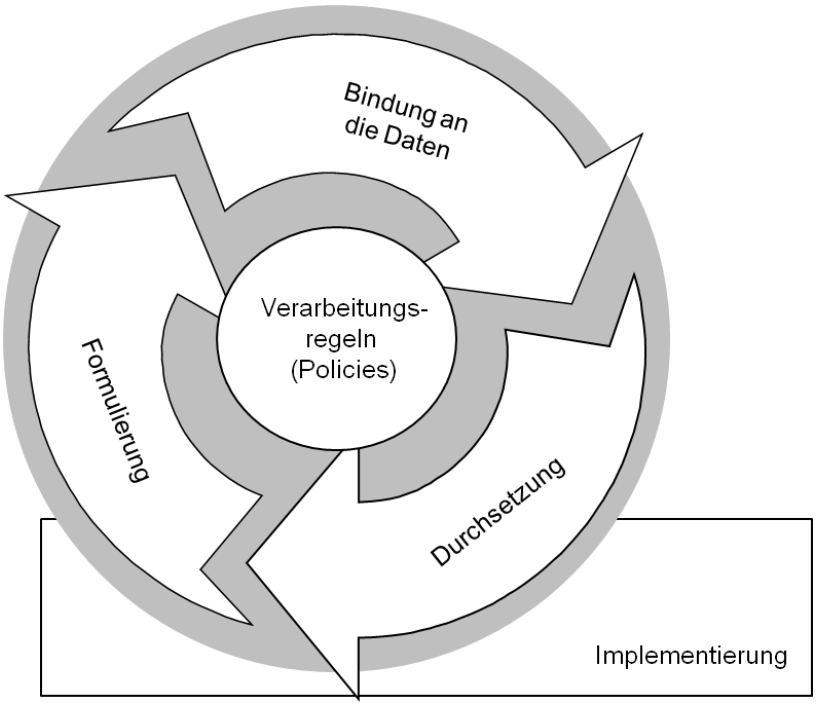


Abbildung 3: Dimensionen von Datenschutzregeln

[26] „This concept of protecting data rather than devices is known as data-centric security.“ [Mark08]
[27] „[…] holistic approach to protecting data assets by assessing the value of individual pieces of information and then defining specific controls and security measures for each. “ [BCSS+06]
[28] „[…] shouldn’t be considered a replacement […] but rather an overlay […]“ [StPV08]

2.3.1 Policies

> *Policies* sind Regeln, mit denen die zur Wahl stehenden Verhaltensweisen von Akteuren in Systemen festgelegt werden[29]. Datenschutz-Policies beziehen sich dabei auf den Umgang mit personenbezogenen Daten.

Datenschutz-Policies begegnen den Nutzern von Online-Diensten häufig in Form der für den Webseiten-Betreiber verpflichtenden Datenschutzerklärung, einem juristischen, natürlichsprachlich formulierten Dokument, das beschreibt, welche personenbezogenen Daten der Betreiber zu welchem Zweck erheben und verarbeiten will, wo die Daten verarbeitet werden und welche Rechte und Eingriffsmöglichkeiten dem Betroffenen demgegenüber zugestanden werden[30]. Der Natur ihrer Formulierung folgend weisen Datenschutzerklärungen dieser Art keine Verknüpfung mit der organisatorischen oder technischen Umsetzung der in ihrem Inhalt festgelegten Regeln auf. Anders gesagt, eine Änderung des Policy-Textes bewirkt für sich genommen keine Änderung im Verhalten des Systems oder in der Auswahl von Handlungsoptionen, die den Akteuren zur Verfügung stehen. Im Gegenzug bewirkt eine Änderung in der Verarbeitungslogik des Systems auch keine automatische Anpassung der Policy.

Der Abschnitt 2.3.2 wird Methoden beschreiben, Policies in maschinenlesbarer Form zu formulieren. Systeme, die entsprechende Module zur Interpretation und Durchsetzung der Policies implementiert haben, können dann unmittelbar auf Policy-Änderungen reagieren und die Verhaltensoptionen anpassen.

Unabhängig von der Art ihrer Beschreibung lassen sich Policies dahingehend unterscheiden, wo sie formuliert werden.

Formulierungsort der Policies:

- **Beim Betroffenen**

  Betroffene nehmen selbst die Definition ihrer Datenschutz-Policies vor. Häufig geschieht dies in nicht-formaler Form, das heißt weder schriftlich noch in einer formalisierten Sprache. Die Entscheidungen fallen häufig spontan, beispielsweise ob und in welcher Ausprägung man an einer Telefonumfrage teilnimmt oder wie man auf Aufforderungen zur Datenerfassung auf Webseiten reagiert. Andere Arten der Formulierung mögen zunächst als dem Betroffenen überlassen erscheinen, stellen sich bei näherer Betrachtung jedoch als solche heraus, bei denen eine

[29] vgl. [Pond02]

[30] z.B. http://www.uni-regensburg.de/datenschutz/index.html oder http://www.amazon.de/gp/help/customer/display.html/ref=footer_privacy?ie=UTF8&nodeId=3312401 (Zugriffe am 25.12.2011).

Verhandlung zwischen Betroffenem und Verarbeiter stattfindet, so etwa bei der Auswahl von Datenschutz-Einstellungen in einem sozialen Netzwerk. Hier kann der Betroffene zwar vermeintlich frei über die Einstellungen zu seiner Privatsphäre entscheiden, aber seine Wahl hat Einfluss auf Umfang und Qualität der ihm zur Verfügung gestellten Dienste. In diesem Sinne ist hier die Freiheit der Wahl eingeschränkt.

- **Beim Verarbeiter**

  Der Kontrollbereich des Verarbeiters kann wohl als der am häufigsten auftretende Formulierungsort für Policies verstanden werden. Nicht nur sind viele Verarbeiter als Betreiber von Webseiten und Applikationen dazu verpflichtet, eine allgemeingültige Datenschutzerklärung gegenüber ihren Kunden vorzuhalten, sie sind im Gegensatz zu diesen auch meist in der Lage, juristisch korrekte Policies zu erstellen und sie den wechselnden Rahmenbedingungen anzupassen. Darin besteht freilich auch die Kehrseite dieses für den Betroffenen bequemen Service: Er muss regelmäßig damit rechnen, dass der Verarbeiter aufgrund des bestehenden Macht- und Wissensgefälles die Policies zu seinen eigenen Gunsten formulieren wird.

- **Bei dritten Parteien**

  Die Vorgabe von Policies durch Dritte (z.B. Hersteller von Webbrowsern, Standardisierungsgremien) ist eher als Vorlage zu finden, die von Betroffenen und Verarbeitern als Ausgangspunkt für die Erstellung ihrer eigenen Policies genutzt wird. Auch vom Gesetzgeber unverrückbar festgelegte Bedingungen bedürfen einer auf den konkreten Anwendungskontext abgestellten Spezifizierung. Immerhin stellen vorbelegte Standard-Policies eine Rückfall-Lösung dar, auf die man sich bedarfsweise berufen kann, wenn Betroffener und Verarbeiter keine konkreteren Regelungen vereinbart haben.

- **Verhandlung zwischen Betroffenem und Verarbeiter**

  Wie zuvor geschildert, sind einige zunächst als freie Wahl des Betroffenen erscheinende Festlegungen in Wahrheit Verhandlungen mit dem Verarbeiter, selbst wenn dieser nicht unmittelbar als Verhandlungspartner auftritt. Er nimmt seinen Part vielmehr durch Vorgabe von Optionen und deren Konsequenzen im von ihm festgelegten Rahmen wahr. Echte Verhandlungen sind allgemein aufgrund des Macht- und Wissensgefälles nur möglich, wenn die Betroffenen sich zu Interessenvertretungen formieren oder sich anderweitig von starken Lobbys vertreten lassen.

Eng mit dem Formulierungsort von Policies verknüpft ist natürlich die Perspektive, aus der sie formuliert sind. Kolter schlägt eine Einordnung der Policies nach 4 Kategorien

vor[31]. Er verwendet dabei die Beziehung zwischen dem Autor und dem Zielpublikum einer Policy als Kriterium:

- **Datenschutzpräferenzen des Dateneigners** (Formulierungsort: Beim Betroffenen, bei Dritten oder Verhandlung)
- **Datenschutzerklärung des Verarbeiters** gegenüber den Betroffenen (Formulierungsort: Beim Verarbeiter, bei Dritten oder Verhandlung)
- **Interne Policies des Verarbeiters** (Formulierungsort: Beim Verarbeiter, bei Dritten oder Verhandlung)
- **Datenschutzauflagen für sekundäre Nutzer**, die Daten vom ursprünglichen Verarbeiter erhalten (Formulierungsort: Beim Betroffenen, beim Verarbeiter, bei Dritten, durch Verhandlung von Betroffenem und Verarbeiter, durch Verhandlung von Verarbeiter und sekundärem Nutzer oder durch Verhandlung aller drei Parteien)

Policies unterliegen wie auch die Objekte, zu deren Schutz sie formuliert sind, einem eigenen Lebenszyklus. Sie müssen initial erstellt, für neue Gegebenheiten angepasst[32] und schließlich außer Kraft gesetzt werden. Skogsrud, Benatallah und Casati beschreiben vier Optionen, wie mit Policy-Änderungen verfahren werden kann, wenn diese während bestehender Kommunikationsbeziehungen stattfinden[33]:

- **Fortführen der Beziehung mit den alten Policies**

  Dies bedeutet, dass das System unter Umständen mit mehreren verschiedenen Policies zur selben Zeit umgehen muss. Es gewährt aber den Beteiligten die Sicherheit, dass einmal Vereinbartes von Bestand ist („Pacta sunt servanda“)[34].

- **Abbrechen der bestehenden Beziehung**

  Ist das Fortführen der Beziehung unter den alten Policies nicht möglich, etwa weil eine Gesetzesänderung dem entgegensteht, ist der Abbruch der Beziehung die einfachste, wenn auch nicht immer zielführende Option. Soll die Beziehung mit neuen Policies wieder belebt werden, ist das möglicherweise mit zusätzlichem Aufwand verbunden.

- **Migration auf die neue Policy**

  Die bisherigen Policies werden durch die neuen Regelungen abgelöst, wobei bereits aktive oder vergangene Transaktionen angepasst werden. Das bedeutet beispielsweise, dass ein Verarbeiter die gemäß früherer Policies vereinbarte

[31] vgl. [Kolt10] in Erweiterung der Definition von Kumaraguru et al. [KCLC07], die nur die Datenschutzpräferenzen des Betroffenen und die Datenschutzerklärungen von Verarbeitern unterscheiden.

[32] Die Anpassungen können sowohl auf technischer Evolution der Plattform, des Schutzgegenstandes oder auch der Angreifermodelle beruhen, wie auch auf Änderungen der äußeren Gegebenheiten wie der Präferenzen Betroffener und Verarbeiter, gesetzlichen Grundlagen oder abzubildenden Organisationsformen.

[33] vgl. [SkBC04]

[34] lat.: „Verträge sind einzuhalten.“

Speicherung bestimmter Daten aufgeben muss (Rollback), sofern die neuen Policies deren Speicherung nicht mehr erlauben.

- **Migration auf eine Hybrid-Policy**

  Eine Migration, jedoch ohne Rollback, ermöglicht diese Option. Die Überführung der bisherigen in die neuen Policies erfolgt unter einer Übergangsregelung, die zeitpunkts- oder kontextbezogen definiert, welchen Festlegungen eine Transaktion unterliegt.

2.3.2 Beschreibung der Policies

Die formalisierte Darstellung von Policies ist vorteilhaft. Sie kann von Rechnern gelesen und interpretiert werden, was eine Voraussetzung für ihre technische Durchsetzung ist. Anders herum können Rechner solche Policies auch selbst nach entsprechenden Vorgaben erstellen, etwa nach einer Auswahl von Optionen in einer Selektionsmaske. Eine solche Erfassungshilfe mindert auch den Nachteil, der Menschen gegebenenfalls dadurch entsteht, dass Policies nicht in ihrer natürlichen Sprache sondern in maschinenlesbarer Form bestehen und überarbeitet werden. Weiterhin eliminieren strukturiert formulierte Policies im besten Fall die Ungenauigkeiten, die eine natürlichsprachliche Fassung mit sich bringt, und sie sind unter Beibehaltung ihrer Semantik auf andere Sprach- und Kulturräume übertragbar[35].

Sprache ist ein Zeichensystem, das aus einer Menge von Zeichen mit jeweils eindeutiger Semantik (im Folgenden: Vokabular) besteht, die mittels grammatikalischer Regeln (Syntax) kombiniert werden, um Aussagen zu treffen[36]. Im hier diskutierten Rahmen ist die Policy die gewünschte Aussage, das Vokabular sind die Elemente, die benötigt werden um Datenschutzregeln eindeutig zu beschreiben, und die Grammatik stellt die Menge der Festlegungen dar, nach denen Policies maschinenlesbar formuliert werden können.

Nun kann eine Policy in einer proprietären Sprache verfasst werden, also einer Sprache, die nur im Kontext einer Applikation, eines Systems oder einer Gruppe von Betroffenen und Verarbeitern Gültigkeit besitzt. Parallel dazu liegen, ausgehend von den Bemühungen einer einheitlichen Beschreibung von Zugriffskontrollen, Vorschläge für allgemein verwendbare Sprachen zur Policy-Beschreibung vor.

Eine umfassende Übersicht bietet Hansen[37] durch Zusammenfassung der Studien von Kumaraguru et al., Madsen et al. und Anderson[38]. Die Autoren identifizieren vier Kategorien nach dem Nutzungsmodell der Sprachen, die sich, wie in Tabelle 1 gezeigt, wieder

[35] Sofiotis beschreibt die Schwierigkeit, Datenschutzregelungen juristisch übertragbar auf europäischer Ebene zu gestalten, vgl. [Sofi04].
[36] Frei nach: http://de.wikipedia.org/wiki/Sprache (Zugriff: 26.12.2011).
[37] Kategorisierung der Sprachen aus [Hans08] in Teilen frei aus dem Englischen.
[38] [KCLC07], [MaCW06], [Ande06]

danach unterscheiden lassen, ob sie für die Policy-Definition aus Sicht des Verarbeiters oder des Betroffenen geeignet sind.

Kategorie	Perspektive	Ausgewählte Beispiele
Fortgeschrittene Sprachen zur Zugangs- und Zugriffskontrolle	Verarbeiter	ODRL, XrML, SAML, WSPL, XACML
	Betroffener	WSPL, XACML
Sprachen für organisationsspezifische Datenschutz-Policies	Verarbeiter	EPAL, DPAL, E-P3P, PRML
Sprachen für Datenschutz-Policies im Web	Verarbeiter	P3P
	Betroffener	APPEL, XPref
Kontext-bezogene Sprachen	Verarbeiter	Geo-Priv, PeerTrust, Protune, Ponder
	Betroffener	Geo-Priv, PeerTrust, Protune

Tabelle 1: Kategorisierung von Policy-Beschreibungssprachen (nach Hansen)

- **Fortgeschrittene Sprachen zur Zugangs- und Zugriffskontrolle**

  Diese Sprachen entstammen der Formulierung von Zugangs- und Zugriffskontrollen, vor allem zur rollenbasierten Zugriffskontrolle[39]. Deren ursprünglichen Wirkungsbereich, die Umsetzung von Sicherheits-Policies einer Organisation durch die Systemadministratoren, erweitern sie, um beispielsweise Nutzungsbedingungen für personenbezogene Daten abzubilden.

  Bekannte Beispiele sind die Security Assertion Markup Language (SAML)[40], die als XML-basierter Standard insbesondere die Portabilität von Zugangs- und Berechtigungsdaten und damit den systemübergreifenden Austausch von Policies fördert, ebenso wie die Extensible Access Control Markup Language (XACML)[41] und ihre Untermengen[42], die sich unter anderem dazu eignen, als gemeinsame Sprache bei der Übermittlung und Aushandlung von Policies zwischen proprietärsprachlichen Datenschutzverfahren zu dienen.

[39] Role Based Access Control – RBAC, vgl. [ANSI04].
[40] http://saml.xml.org/saml-specifications (Zugriff: 21.03.2012).
[41] [GuBh05], [Ande05]
[42] wie die Web Services Policy Language (WSPL), vgl. [Ande04].

Eine interessante Variante bilden Sprachen, die aus dem Digital Rights Management kommend, in den Dienst des Datenschutzes gestellt werden können. Diese Rights Expression Languages (REL), zum Beispiel die Open Digital Rights Language (ODRL) oder die Extensible Rights Markup Language (XrML) arbeiten mit Datenlizenzen und Befehlen zur Nutzungskontrolle der Daten an Empfängergeräte[43].

- **Sprachen für organisationsspezifische Datenschutz-Policies**

  Sie folgen demselben Konzept wie die vorgenannten Sprachen, wurden jedoch im Gegensatz zu diesen konkret für die Verwendung zum Datenschutz entwickelt. Dementsprechend tritt hier der Zugriff auf Daten gegenüber dem bei klassischer Zugriffskontrolle stärker fokussierten Zugriff auf Funktionen in den Vordergrund. Sprachen dieser Kategorie werden von Organisationen genutzt, um ihre intern formulierten Datenschutz-Policies umzusetzen.

  Die Platform for Enterprise Privacy Practices (E-P3P)[44] und ihr Nachfolger, die Enterprise Privacy Authorization Language (EPAL)[45] erlauben Verarbeitern, den Fluss von personenbezogenen Daten innerhalb ihrer Organisation und den Umgang mit diesen Daten an den veröffentlichten Datenschutzerklärungen auszurichten. EPAL stellt Sprachelemente zur Verfügung, mit denen definierte Aktivitäten für Datenkategorien und unterschiedliche Betroffenengruppen für konkrete Verarbeitungszwecke erlaubt oder verboten werden. Auch das Binden von Verpflichtungen an datenbezogene Aktivitäten ist möglich[46]. Möller schließt im Vertrauen auf die solchermaßen innerorganisatorisch umgesetzte Zugriffssteuerung: „So wird für den Bereich der automatisiert überprüfbaren Datenschutzregelungen rechtskonformes Verhalten des Unternehmens und der Schutz des informationellen Selbstbestimmungsrechts der Kunden und Arbeitnehmer sichergestellt.“[47]. EPAL Policies lassen sich alternativ in der erweiterten Declarative Privacy Authorization Language (DPAL)[48] abbilden und organisationsübergreifend verknüpfen.

  Ihr verwandt ist die Privacy Rights Markup Language (PRML), deren Policies sich aus den Sprachelementen Rolle, Operation, Datenkategorie, Betroffener, Zweck, Beschränkung, Aktion und Transformation zusammensetzen lassen. Hervorzuheben sind die Möglichkeiten, Bedingungen für Aktionen zu setzen, und damit komplexe Policies mit Ereignissteuerung zu realisieren[49].

[43] vgl. [FrKA04], [BeGü04], [Nütz05], [Grim05]
[44] [AHKS02], [KaSW02]
[45] [AHKP+03], [BaDS04]
[46] Beispielsweise wird die Verpflichtung, ein Datum nach Ablauf einer festgelegten Frist zu löschen, an die Aktion zur Speicherung des Datums gebunden. Zu den Elementen von EPAL vgl. [Möll04].
[47] [Möll06]
[48] [BaMR04]
[49] vgl. [Hans08]

- **Sprachen für Datenschutz-Policies im Web**

  Hierbei handelt es sich um Sprachen zur standardisierten Darstellung von Datenschutz-Policies für Dienste (server-seitig) und Datenschutz-Präferenzen der Betroffenen (client-seitig), die sowohl von Menschen gelesen als auch von Rechnern interpretiert werden können. Sie sind daher im Gegensatz zu den vorgenannten Kategorien insbesondere zur Aushandlung von Nutzungsbedingungen zwischen den beteiligten Parteien geeignet.

  Die Platform for Privacy Preferences (P3P)[50] hat sich als Standard dafür etabliert, die Datenschutz-Policies von Webseiten zu formulieren. P3P-Policies sind im Prinzip XML-basierte Fragebögen, in denen der Serviceanbieter standardisierte Aussagen zu den auf dem Server gespeicherten Daten, deren vorgesehener Verwendung, der Aufbewahrungszeit und den Rechten des Betroffenen gibt[51]. Web-Browser können die P3P-Policies der Server lesen und dem Anwender darstellen, sowie sie gegen dessen eingestellte Präferenzen verproben.

  Ausgehend von P3P haben sich „A P3P Preference Exchange Language“ (APPEL)[52] und mit Verbesserungen in der Handhabung die „XPath-based Preference Language“ (XPref)[53] gebildet, mit deren Hilfe sich die Präferenzen des Nutzers applikationsunabhängig formulieren lassen.

- **Kontext-bezogene Sprachen**

  Diese Sprachen berücksichtigen den (auch semantischen) Anwendungskontext bei der Interpretation von Datenschutz-Policies und eignen sich dadurch besonders für personalisierte oder ortsbasierte Services. Diese Kategorie ist im Vergleich relativ jung und es haben sich noch keine Standards etabliert.

  Geo-Priv[54] beispielsweise („Geographic Location/Privacy“) fokussiert auf ortsbezogene Dienste und besitzt Attribute zur Festlegung, wie Informationen über den Aufenthaltsort des Betroffenen übermittelt und wie stark sie gegebenenfalls dabei verfremdet werden.

  PeerTrust[55] und Protune[56] zielen spezifisch auf das maschinelle Aushandeln von Vertrauensbeziehungen zwischen Parteien, wobei Zertifikate, Regeln, aber auch Metadaten über die zu schützenden Objekte ausgetauscht und dynamisch in der Verhandlung berücksichtigt werden.

[50] http://www.w3.org/P3P/ (Zugriff: 21.03.2012).
[51] [HaAb04], [Feda07] diskutieren die Vor- und Nachteile der Festlegung in P3P auf einen verhältnismäßig kleinen Satz von Standard-Antworten.
[52] http://www.w3.org/TR/P3P-preferences/ (Zugriff: 21.03.2012).
[53] [AKSX05]
[54] vgl. [CoHa09], [STMC+07]
[55] vgl. [GNOS+04]
[56] [BoOl05], [BDOS08]

Ponder[57] als letztes Beispiel verfügt über die Mittel, organisatorische Strukturen in seinen Policies abzubilden und so Regeln und Rollen miteinander in Beziehung zu setzen. Besonders in großen, verteilten Strukturen kann es so herangezogen werden, um basierend auf einem übersichtlichen Satz an grundlegenden Policies („primitive policies“) und der Gruppierung sowie Verknüpfung von Objekten („domains“) ein mit der Struktur mitwachsendes Policy-Werk zu realisieren.

Suchte man nun im Sinne des eingangs geforderten Selbstdatenschutzes Kandidaten, die zur Formulierung von Datenschutz-Policies aus der Perspektive des Betroffenen geeignet sind, und die sich für den allgemein gültigen Einsatz eignen, also auch von Nicht-Experten sicher bedient werden können, bleiben aus der obigen Auswahl mit Einschränkungen APPEL und XPref mit erweitertem P3P-Vokabular, oder XACML-Subsets.

2.3.3 Bindung an die Daten

Wer Policies zum Schutz seiner personenbezogenen Daten formuliert, hat ein Interesse daran, die Policies auch durchzusetzen. Die Qualität der Durchsetzbarkeit hängt wiederum davon ab, wie verbindlich die Bindung ist, die zwischen den Daten und ihren Policies besteht.

Die schwächste Form der Bindung ist die rein **organisatorische**. Betroffener und Verarbeiter vereinbaren die Gültigkeit einer Policy für ein bereitzustellendes Datum. Vermutet der Betroffene später, dass seine Daten missbräuchlich, also entgegen den Bestimmungen der Policy abgerufen und verwendet wurden, muss er zunächst detektivisch die Nutzung seiner Daten beim Verarbeiter nachvollziehen, dann den Verstoß gegen die Policy dokumentieren, anschließend beweisen, dass sich beide Parteien der selben Version der Policy verpflichtet haben und schließlich juristische Sanktionen gegen den Verarbeiter erwirken.

Eine Verbesserung ist eine Bindung von Policy und Daten in der Form, dass Policy-Verstöße direkt **nachweisbar** werden. Dies bedingt eine vollständige Protokollierung aller Datenabrufe durch den Verarbeiter – und der bei ihm stattfindenden Verwertungsschritte, sofern man ein solches Maß an Kontrolle über die interne IT-Landschaft des Verarbeiters erlangen könnte – und eine Zugänglichkeit der Protokolldateien für den Betroffenen. Eine Steigerung ist der **automatische Nachweis** von Verstößen gegen die Policy. Dazu ist neben der Protokollierung auch ein Alarmsystem notwendig, das Verstöße erkennt, bewertet und meldet.

Die vorgenannten Arten der Policy-Bindung wirken vor allem vor dem Hintergrund juristischer Sanktionierung von Verstößen, da sie auf nachgelagerte (detektivische) Kontrollen ausgerichtet sind. Datenschutzfreundliche Technik soll aber präventiv wirken, also Missbrauch verhindern, und erst bei Misslingen dieser Aufgabe geeignete

[57] [DDLS01], [Pond02]

Maßnahmen für die detektivische Arbeit bereitstellen. Ziel muss es folglich sein, eine **technisch nicht umgehbare** Bindung von Policies an ihre Daten zu erreichen.

Das bislang beste Konzept, technische Durchsetzbarkeit mit juristischer Verwertbarkeit zu kombinieren, ist das Paradigma der „**Sticky Policies**“, die den Daten, auf die sie sich beziehen, anhaften. Meints formuliert: „Kernidee dieses Konzeptes ist es, personenbezogene Daten stets mit einer ausgehandelten Datenschutz-Policy zu versehen und diese auch bei Datenübermittlung an Dritte den Daten weiter anhaften zu lassen.“[58]

Da Sticky Policies spezifischen Datenpaketen anhaften, sind sie individuell gestaltbar und somit auf den konkreten Schutzbedarf des Betroffenen und der Datenkategorie abgestellt. Diese Individualität bedeutet erhöhten Aufwand bei der Policy-Erstellung und -Pflege, lässt dem Betroffenen aber im Gegenzug die Hoheit über seine Präferenzen[59].

Casassa Mont, Pearson und Bramhall nennen zwei äquivalente Möglichkeiten, Policies physisch an die entsprechenden Daten zu binden[60]:

- **Public Key Kryptographie**

  Die zu schützenden Daten und die dazu gehörende Policy werden mit einem symmetrischen Schlüssel verschlüsselt. Dieser Schlüssel und ein Hashwert der Policy werden gemeinsam mit dem öffentlichen Schlüssel eines Datenschutz-Managementsystems verschlüsselt. Das verschlüsselte Paket, die symmetrisch verschlüsselten personenbezogenen Daten und die Policy werden gemeinsam gespeichert. Erzeugt ein Anforderer mit dem öffentlichen Schlüssel des Systems und der von ihm anerkannten Policy ein übereinstimmendes Policy-Chiffrat, gibt das System den symmetrischen Schlüssel an den Anforderer frei, den dieser anschließend zur Entschlüsselung der Daten verwendet.

- **Identity-based Encryption (IBE)**[61]

  Hier ist die Policy bzw. ein aus ihr generierter Hashwert ein Teil des Geheimnisses, das benötigt wird, die verschlüsselten personenbezogenen Daten zu entschlüsseln. Verarbeiter können also nur Anfragen bezüglich der betreffenden Daten stellen, wenn sie die Policies anerkennen. Dies bringen sie nach erfolgreicher Übermittlung des Chiffrats mit der Entschlüsselung zum Ausdruck, da die Anwendung einer verfälschten Policy den Schlüssel verändern und damit die Entschlüsselung scheitern lassen würde.

[58] [Mein06]
[59] vgl. [AsPS02]
[60] vgl. [CaPB04]
[61] Erstmals vorgeschlagen von Adi Shamir [Sham84], erste vollständig funktionale Algorithmen basierend auf quadratischen Residuen von Cocks [Cock01] und mit Hilfe der Weil-Paarung auf elliptischen Kurven von Boneh und Franklin [BoFr01]. Von Holt et al. erweitert um die Möglichkeit, zusätzliche Attribute („Hidden Credentials“) in den Schlüssel einzubringen, vgl. [HBSO03], [BrHS04]. Das Unternehmen Voltage Security setzt etwa in seinen Produkten zur E-Mail-Verschlüsselung und zum gesicherten Zahlungsverkehr auf IBE, s. http://www.voltage.com/products (Zugriff: 17.05.2012).

2.3.4 Durchsetzung

Wo und wann Policies durchgesetzt werden, ist natürlich stark mit dem zugrunde liegenden Datenschutzkonzept verknüpft. Grundsätzlich bestehen die in Tabelle 2 dargestellten Kombinationsmöglichkeiten. Die Durchsetzung der Policies kann bereits **beim Betroffenen** erfolgen. Dabei entscheidet er individuell über die Anfragen von Verarbeitern. Wegen der Möglichkeit der Einzelfallentscheidung kann er sogar auf das Schreiben formaler Policies verzichten, sofern er den Daten keine zusätzlichen Verpflichtungen mitgeben möchte. Die Policy-Durchsetzung erfolgt also **präventiv** und ist der Datenübermittlung **vorgelagert**. Bei Vorliegen eindeutiger Policies kann der Betroffene die Entscheidung über die Datenherausgabe auch vertrauenswürdigen **Dritten** überlassen, bei denen er seine Daten zwischenlagert. Das ist vorteilhaft, wenn der Betroffene für den Zeitpunkt eventuell dringlicher Datenanfragen keine Verfügbarkeit der eigenen Person oder des eigenen Systems garantieren kann. Zugleich kann der Dritte bei geeigneter Aufstellung als Zeuge oder Vermittler im Streitfall dienen. Im Gegenzug dafür gibt der Betroffene die Möglichkeit der Entscheidung im Einzelfall auf.

Ist die Durchsetzung der Policies **zum Verarbeiter** verlagert, kann sie natürlich erst **nach Übermittlung** der Daten zu diesem stattfinden. Entweder findet die Policy-Prüfung vor der gewünschten Verarbeitung der gespeicherten Daten (**präventiv**) statt oder **nachgelagert**, zum Nachweis der Policy-Einhaltung oder Aufdeckung von Verstößen. Die Kombination beider Varianten entspricht der gängigen Praxis bei den aktuellen Personalverwaltungssystemen – Umsetzung der Policies durch Zugangs- und Zugriffskontrolle, sowie nachgelagerte Audits der stattgefundenen Zugriffe gegen die Datenschutzerklärung[62]. In jedem Fall erfordert die Policy-Durchsetzung beim Verarbeiter ein hinreichendes Vertrauen des Betroffenen in die Redlichkeit des Verarbeiters oder zumindest in die Vertrauenswürdigkeit dessen IT-Landschaft[63].

Wurde die Datenhaltung durch den Betroffenen **an einen Dritten** ausgelagert, ist es auch denkbar, dass Teile der Verarbeitung dieser Daten (z.B. Aggregation, Filtern, Transformation) im Auftrag des Verarbeiters von dem Dritten durchgeführt werden. In diesem Sinne kann der Dritte ebenfalls die Datenschutz-Policies **vor der Übermittlung** der resultierenden Daten an den Verarbeiter prüfen beziehungsweise später den **detektivischen Nachweis** der Verarbeitung erbringen.

In einem weiteren möglichen Szenario wird auf eine echte, vor- oder nachgelagerte, Durchsetzung von Vereinbarungen verzichtet. An deren Stelle kann etwa die soziale Kontrolle eines Reputationssystems treten, das Policy-Verstöße indirekt durch Entzug von Vertrauen einer Gemeinschaft sanktioniert. Diese Option wird jedoch regelmäßig nicht für Daten präferiert sein, die von hoher Sensibilität für die Beteiligten sind.

[62] vgl. u.a. [LeOS11]

[63] Die Möglichkeiten, mittels Techniken des Digital Rights Management und Trusted Platforms Datenschutz-Policies beim Verarbeiter zu erzwingen, wird im weiteren Verlauf dieser Arbeit noch untersucht.

Zeitpunkt \ Ort	Beim Betroffenen	Beim Verarbeiter	Bei Dritten
Präventiv, vor der Übermittlung	●		●
Präventiv, vor der Verarbeitung		●	
Detektivisch, nach der Verarbeitung		●	●
Nur Willenserklärung	○	○	○

Tabelle 2: Ort und Zeitpunkt der Durchsetzung von Policies

Die Durchsetzung von Datenschutz bedarf nicht nur der Beachtung im unmittelbaren Kontext der Nutzung durch einen Verarbeiter. Ebenso sind die anderen Phasen im Lebenszyklus der zu schützenden Daten durch datenschutzfreundliche Techniken zu unterstützen. Abhängig vom Untersuchungsziel lässt sich der Lebenszyklus ganz unterschiedlich darstellen. Im hier diskutierten Umfeld soll eine Detaillierung gemäß Abbildung 4 dazu genutzt werden, die wesentlichen Schritte zu betrachten.

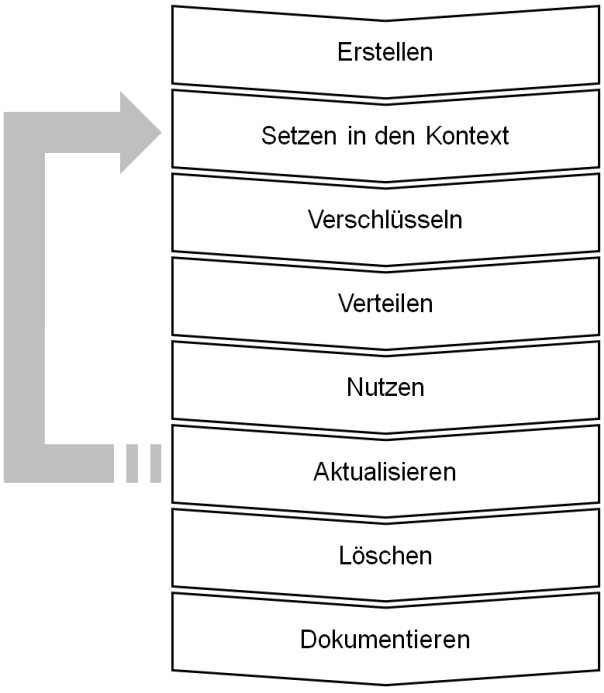


Abbildung 4: Lebenszyklus von Daten[64]

[64] Frei nach [Baha10], [BSI11].

- **Erstellen**

  Ein System, das den Betroffenen dabei unterstützt, seine Daten in geeigneter Form zu formulieren, kann ihn bei der Beachtung der Datensparsamkeit im Sinne des Selbstdatenschutzes begleiten. Zugleich kann hier bereits die formale Einordnung der Daten in der Form erfolgen, dass anschließend eindeutige Policies definierbar sind.

- **Setzen in den Kontext**

  Das Setzen in den Kontext bedeutet die Abstimmung der Daten mit den sie umgebenden Systemen. Bei Systemen, die Sticky Policies, also eine feste Bindung von Policies an Daten, vorsehen, nimmt dieser Vorgang einen wichtigen Teil dieses Schritts ein. Neben der Policy-Bindung kann dieser Schritt aber auch weitere Aufgaben enthalten, die zur Vorbereitung der technischen Ablage benötigt werden, beispielsweise das Übersetzen der Daten in ein maschinenlesbares Format.

- **Verschlüsseln**

  Häufig erfolgt die Datenübermittlung und -ablage nicht im Klartext, sondern insbesondere zum Schutz der Vertraulichkeit in verschlüsselter Form. Die Anwendung einer vorab festgelegten Chiffrierung auf die betroffenen Daten erfolgt in diesem Schritt. Zusätzlich zur begleitenden Sicherung der Kommunikation kann die Verschlüsselung auch ein innerer Bestandteil des Systems sein, wie es bei kryptographischen Containern[65] der Fall ist. Hier nimmt die Verschlüsselung eine zentrale Aufgabe im Schutz der Daten vor dem Kommunikationspartner selbst ein.

- **Verteilen**

  Die logistische Aufgabe, ob, wann und in welcher Form die betroffenen Daten verteilt werden, kann unterschiedliche Ausprägungen annehmen. Bei einer exklusiven Speicherung von Daten beim Betroffenen liegt die Verteilung ganz am Ende eines jeden Protokolls, erfolgt strikt anlassbezogen und stets auf unmittelbare Genehmigung des Betroffenen hin. Komplexer wir die Verteilung der Daten in Systemen, die zum Zeitpunkt der Datenaufdeckung nicht die Möglichkeiten zur Interaktion zwischen Betroffenem und Verarbeiter vorsehen.

- **Nutzen**

  Die Nutzung ist sicher der vielschichtigste Aspekt im Lebenszyklus eines Datensatzes. Und in gleichem Maße ist ihre Kontrolle eine Aufgabe hoher Komplexität. Sie erfordert vor allem Eingriffe in den Einflussbereich des Verarbeiters.

[65] vgl. [IBM96]

- **Aktualisieren**

  Daten unterliegen dem Verfall und eine gute datenschutzfreundliche Technik muss Optionen zulassen, Datensätze ebenso wie Policies über die Zeit anzupassen, wenn sich die Bedingungen oder Inhalte ändern. Aktualisierungen können die Notwendigkeit nach sich ziehen, wieder zurück zum Schritt der Kontextualisierung zu gehen, sowie die Verschlüsselung und Verteilung erneut durchzuführen.

- **Löschen**

  Eine besondere Art der Aktualisierung ist die Löschung von Datensätzen. Sie bildet den Abschluss des Lebenszyklus eines jeden Datensatzes. Im Gegensatz zum Verblassen oder Vergessen werden ist das Löschen von Daten ein bewusster Akt, der von einem Verantwortlichen unter Kenntnisnahme mindestens des Betroffenen zu einem definierten Zeitpunkt erfolgt. Die Qualität der Löschung hängt davon ab, in welchem Maße Kopien der Daten sowie physische Restspuren in den Löschvorgang zuverlässig und vollständig einbezogen werden.

- **Dokumentieren**

  Der Lebenszyklus der Daten selbst, von der ersten strukturierten Anlage eines Datensatzes, über die Übermittlung und Nutzung, bis hin zur Löschung ist im besten Fall lückenlos dokumentiert und von den Berechtigten les- und interpretierbar. Dabei ist darauf zu achten, dass für die Dokumentation selbst ebenfalls Regelungen gelten, was ihren Schutz vor unbefugter Nutzung, ihre Formulierung und ihren eigenen Lebenszyklus angeht.

2.3.5 Technische Implementierung

Abgeleitet von Kurkovsky et al. lassen sich die Umsetzungen von datenschutzwirksamen Systemen nach der Implementierung einer oder mehrerer der folgenden Techniken zum Datenschutzmanagement unterteilen[66]:

- **Zugangskontrolle** (Access Rights Management)

  Nur identifizierte und autorisierte Subjekte können am Datenaustausch teilnehmen.

- **Zugriffskontrolle** (Access Policy Management)

  Den Teilnehmern können im System unterschiedliche Rechte zugewiesen werden. Im Zusammenwirken mit der Datenklassifizierung werden Zugangs-Policies erstellt.

[66] [KuRB08], Übersetzung der Überschriften durch den Autor; Englische Originalbezeichnungen in Klammern.

- **Klassifizierung der Daten** (Classification of Resources)

  Die personenbezogenen Daten können in verschiedene Kategorien eingeteilt werden, die mit unterschiedlicher Offenheit gegenüber der Herausgabe versehen werden können. Diese Technik wird insbesondere dann wirksam, wenn sie das Formulieren von Policies erlaubt, die den unterschiedlichen Klassen verschiedene Zielgruppen zuordnen.

- **Kontrolle der Datenpersistenz** (Data Persistence Control)

  Abgelegte Daten unterliegen einem definierten Lebenszyklusmodell. Das bedeutet insbesondere, dass sie mit Attributen versehen werden können, die ihre Lebenszeit beschreiben, d.h. den Zeitraum, in dem sie gespeichert, genutzt und weitergegeben werden können. Vollständige Persistenzkontrolle sorgt zudem dafür, dass Daten nach Ablauf dieser Lebenszeit zuverlässig gelöscht werden.

- **Steuerung der Granularität** (Granularity Awareness)

  Für verschiedene Zwecke und Arten der Datenverarbeitung können unterschiedliche Detailebenen der Daten notwendig sein. Kurkovsky et al. nennen beispielhaft lokalisierte Dienste, von denen einige auf meter- und sekundengenaue Angaben zum Aufenthaltsort angewiesen sind, für andere der Maßstab von Kilometern und Stunden jedoch ausreichend ist. Auch abseits der ortsbasierten Dienste ist die Steuerung der Granularität von Datenweitergabe von Relevanz. So ist es nicht für jeden Dienst notwendig, das vollständige Geburtsdatum einer Person zu erfahren. Wenn es um Altersverifikation geht, reicht meist das Geburtsjahr. Oder anstelle der vollständigen Adresse kann in vielen Fällen schon die Angabe der Stadt oder des Stadtteils ausreichend sein.

- **Berechtigungen** (Constraints and Permissions)

  Die auf den abgelegten Daten möglichen Aktionen werden mit Berechtigungen versehen, die sich wiederum zusammenfassen und (beispielsweise im Modell der Role-based Access Control) an verschiedene Verarbeiter zuweisen lassen.

- **Hinterlegung der Eigentümerschaft** (Ownership of Context Information)

  Jedes Datum im System ist mit einem eindeutigen Eigentümer versehen, der bewusste Kontrolle über das Datum ausübt. Die Eigentümerschaft bedeutet jedoch nicht unbedingt, dass sie anderen als dem Eigentümer bekannt sein muss.

- **Verschleierung des Informationsflusses** (Information Flow Obfuscation)

  Der Inhalt und tatsächliche Umfang jeder Kommunikation, die dem Datenfluss vom Eigner zum beabsichtigten Verarbeiter dient, ist gegenüber Dritten verschleiert. Die Gewinnung von Informationen aus der Analyse der Kommunikationskanäle wird erschwert oder verhindert.

- **Schutz des Dienstzugangs** (Service Access Protection)

  Der Betroffene wird davor geschützt, dass seine grundsätzliche Teilnahme an einem Dienst oder Protokoll gegenüber Dritten bekannt wird.

- **Angemessene Datenweitergabe** (Information Disclosure Protection)

  Daten des Betroffenen werden nur in dem Umfang abgerufen und gespeichert, der für die vereinbarte Nutzung des Verarbeiters notwendig ist.

- **Geschützte Verwendung** (Protection of Information Usage)

  Hier steht die zweckgebundene Verwendung der übermittelten Daten im Vordergrund. Die personenbezogenen Daten des Betroffenen werden nur für den vereinbarten Zweck verwendet. Falls sie zur Erfüllung dieses Zwecks über einen längeren Zeitraum durch den Verarbeiter gespeichert werden, wird sichergestellt, dass sie dadurch keinem Dritten zugänglich werden oder durch den Verarbeiter einem anderen als dem vereinbarten Zweck zugeführt werden.

2.3.6 Zusammenfassung der Ordnungskriterien

Mit den vorangehenden Abschnitten steht ein umfangreiches Gerüst für die Einordnung von Verfahren zur technischen Verarbeitungskontrolle zur Verfügung. Die Fragen, die im Design eines Verfahrens beantwortet werden sollten, sind demnach:

- Welcher **Quelle** entstammen die verwendeten Policies und aus welcher **Perspektive** sind sie formuliert?
- Wie sind die Policies **ausgedrückt** und wie werden Policy-Änderungen in deren Lebenszyklus eingebracht?
- Auf welche Weise sind die Policies an die Daten **gebunden**?
- Wo findet die **Durchsetzung** der Policies statt und zu welchem **Zeitpunkt**, bezogen auf die Übermittlung und Verarbeitung der Daten?
- Welche Phasen im **Lebenszyklus** der personenbezogenen Daten deckt das Verfahren ab?
- Welche **Techniken** sind implementiert?

Mit Abbildung 5 wird ein Rahmen vorgestellt, in dem sich die Eigenschaften eines Verfahrens zur Verarbeitungskontrolle kompakt darstellen lassen. Dieser „Steckbrief“ besteht aus einem Abschnitt links oben zur Beschreibung des Policy-Managements, einer Tabelle links unten, die im jeweiligen Verfahren mögliche Orte und Zeitpunkte der Policy-Durchsetzung zeigt, sowie einem Indikator rechts für die Phasen des Datenlebenszyklus, in denen das Verfahren hauptsächlich wirkt.

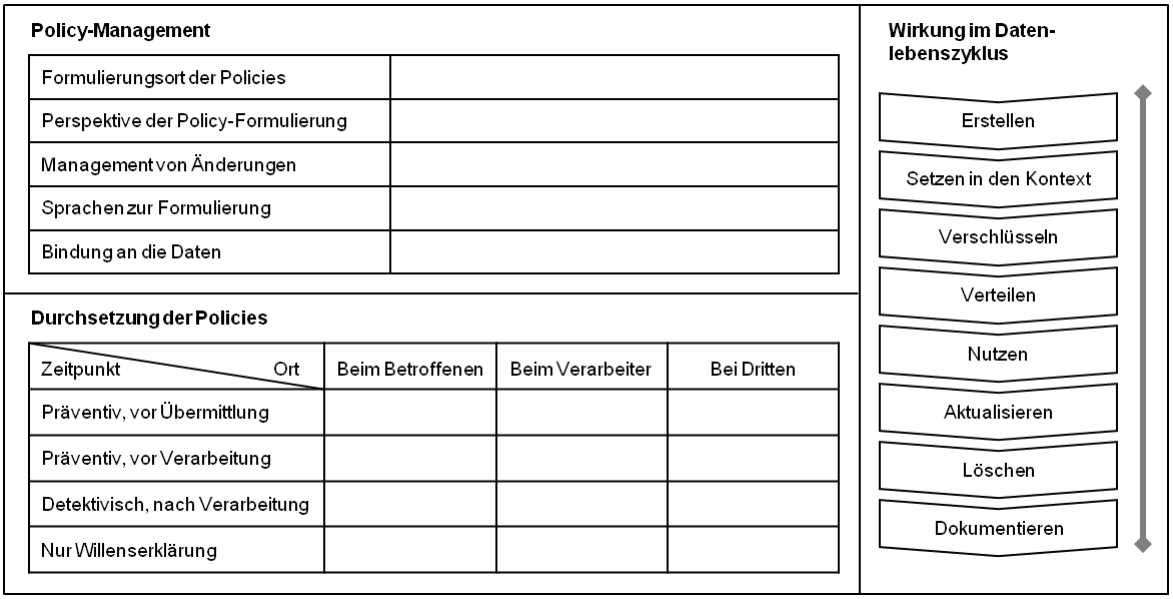


Abbildung 5: Steckbrief für Verfahren zur Verarbeitungskontrolle

Diese Vorlage wird in Kapitel 3 zur Beschreibung und Einordnung der dort vorgestellten Verfahren genutzt. Der Steckbrief enthält keinen Abschnitt für die Darstellung der implementierten Techniken nach Kapitel 2.3.5, da jede davon in allen Verfahren in unterschiedlich gewichteten Kombinationen beteiligt ist.

Sind die Fragen zum Design eines Verfahrens beantwortet, kann seine Bewertung erfolgen. Auch wenn diese immer auf den spezifischen Anforderungen beruhen wird, für die ein Verfahren zum Einsatz gelangen soll, also die Gewichtung einzelner Kriterien dem jeweiligen Anwendungskontext überlassen sein muss, so lassen sich doch unter der Zielsetzung des effektiven, technisch unterstützten Selbstdatenschutzes grundlegende Qualitätsannahmen treffen:

- Ein Verfahren, das **präventive** Maßnahmen zur Verarbeitungskontrolle einsetzt, ist der Nutzung rein nachgelagerter Kontrollen vorzuziehen.
- Eine **Kombination** beider Kontrollarten wiederum stärkt die Effektivität des Gesamtsystems.
- Je weniger **Vertrauen** des Betroffenen in die anderen beteiligten Parteien eines Verfahrens notwendig ist, um dessen Funktionieren zu gewährleisten, umso besser ist das Verfahren[67].
- Stark an die Daten **gebundene Policies** sind unumgänglich, wenn die Policy-Durchsetzung außerhalb des Wirkungsbereichs des Betroffenen stattfindet.
- Starke Policies umfassen einen möglichst **breiten Korridor im Lebenszyklus** der Daten.

[67] vgl. [RaPM96] zu „mehrseitiger Sicherheit“, die eine Berücksichtigung der Sicherheitsanforderungen aller beteiligter Parteien fordert, sie alle aber zugleich auch als potentielle Angreifer des Verfahrens identifiziert.

3 Ausgewählte Verfahren zur Verarbeitungskontrolle

Das Kapitel beschreibt ausgewählte Verfahren, die repräsentativ für verschiedene Ansätze der Verarbeitungskontrolle im Sinne des Datenschutzes stehen. Die vorgestellten Verfahren werden jeweils in einem Steckbrief[68] entlang der in Kapitel 2.3 ausgelegten Ordnungskriterien untersucht.

Dem Ziel umfassenden Selbstdatenschutzes durch Verarbeitungskontrolle kommen Ansätze nahe, deren reale Umsetzung heute an den technischen oder ökonomischen Bedingungen scheitern (Speicherung beim Betroffenen, der die ausschließliche Kontrolle über seine Daten zu jedem Zeitpunkt innehat, sowie das Privacy-DRM, das technisch erzwungene Kontrolle des Betroffenen über die Infrastruktur der Datenverarbeiter voraussetzt). Die realisierten Ansätze demgegenüber beschränken ihren Schutzumfang jeweils auf Teilaspekte des Ziels. Beide Feststellungen werden im Verlauf dieses Kapitels erläutert.

3.1 Speicherung beim Betroffenen

Möchte man datenschutzfreundliche Techniken beurteilen, lohnt zunächst die Betrachtung eines Modells mit idealisierten Aspekten. Ein Datenschutzniveau, das man für dieses Modell nachweisen kann, stellt die Obergrenze für den erzielbaren Datenschutz in Modellen dar, die ohne optimierte Prämissen auskommen müssen. Die exklusive Speicherung beim Betroffenen arbeitet solchermaßen mit optimierten Aspekten.

3.1.1 Voraussetzungen

In dieser Betrachtung wird angenommen, dass personenbezogene Daten exklusiv beim Betroffenen gespeichert werden. Die Speicherung erfolgt auf einem sicheren Gerät, auf das der Betroffene jederzeit uneingeschränkt zugreifen kann. Anderen Personen oder technischen Einrichtungen ist ohne Steuerung durch den Betroffenen keinerlei Zugriff auf das Gerät oder die dort gespeicherten Daten möglich[69]. Mit Zustimmung und Überwachung durch den Betroffenen können Verarbeiter Daten aus dem Gerät auslesen, Daten dort speichern oder zwischen den Parteien vereinbarte Verarbeitungen innerhalb des Gerätes durchführen. Eine Speicherung von Daten des Betroffenen außerhalb des Geräts sei den Datenverarbeitern in diesem Modell nicht möglich, was dem Betroffenen maximale Kontrolle bezüglich der Weitergabe und Nutzung seiner Daten gewährt.

[68] Abbildungen auf den Seiten 40, 46, 53, 61 und 64.

[69] Federrath und Pfitzmann bezeichnen die persönliche Rechenumgebung als den *Vertrauensbereich* des Benutzers, für den angenommen wird, dass Angriffe innerhalb dieses Bereichs nicht stattfinden, vgl. [FePf11].

„Speicherung beim Betroffenen“ versteht sich hier nicht als die Hinterlegung von persistenten Cookies durch Anbieter auf dem Endgerät des Betroffenen, da bei Cookies die Anlage oder das Auslesen eben nicht vollständig unter Kontrolle des Betroffenen erfolgt[70]. Zwar kann typischerweise das Anlegen in Web-Browsern gesteuert werden, der konkret abgelegte Inhalt oder das spätere Auslesen entzieht sich jedoch der Kontrolle des Betroffenen. Löschen ist wiederum durch den Verarbeiter nicht möglich[71] und für den Betroffenen mangels einfacher Identifizierbarkeit schwierig.

Das „sichere Gerät“ zur Speicherung personenbezogener Daten beim Betroffenen ist im Prinzip als Kombination einer Smart Card mit einem Gerät zu verstehen, das eine permanente Internet-Verbindung zur Verfügung hat oder zumindest bei Anfragen jederzeit zu erreichen ist, etwa ein Smartphone.

3.1.2 Speichern der Daten

Zur Speicherung der personenbezogenen Daten stehen im Modell verschiedene Optionen zur Verfügung. Im ersten Ansatz sammelt der Betroffene zu einem Startzeitpunkt alle Daten, zu deren Herausgabe er grundsätzlich bereit ist, wenn bestimmte Bedingungen erfüllt werden. Diese Daten werden zusammen mit den gewünschten Policies im sicheren Gerät gespeichert. Hinzu kommen die Daten, die der Betroffene aufgrund gesetzlicher oder vertraglicher Regelungen zur Verfügung stellt. Die an letztere gebundenen Policies sind nicht durch den Betroffenen zu erstellen, sondern durch den Vertragspartner (beispielsweise den Arbeitgeber des Betroffenen) oder eine öffentliche Stelle in Form von Vorlagen zur Verfügung zu stellen. Alternativ werden zunächst auch keine oder nicht alle Policies hinterlegt. Für den Fall, dass ein konkretes Datum angefragt wird, für das noch keine Regelungen getroffen wurden, richtet das sichere Gerät eine Anfrage an den Betroffenen und gibt erst nach dessen Zustimmung die gewünschten Daten heraus. Die Entscheidung des Betroffenen wird gespeichert, so dass künftige gleich lautende Anfragen desselben Verarbeiters bereits automatisch beantwortet werden können[72]. Der Ansatz eignet sich vorwiegend in Umgebungen, deren Datenbedarf a priori feststeht, weil er beispielsweise Bestandteil eines Vertrages zwischen Betroffenem und Verarbeiter ist.

Steht die Art der benötigten Daten nicht grundsätzlich fest, liegt ein inkrementelles Vorgehen näher. Für jede Anfrage schlägt das sichere Gerät nach, ob die angeforderten Daten bereits gespeichert sind. Falls ja, folgt das weitere Vorgehen dem zuerst geschilderten Ansatz. Falls nicht, fragt das Gerät beim Betroffenen zunächst nach der grundsätzlichen Bereitschaft zur Weitergabe der betreffenden Daten, dann nach gegebenenfalls festzuhaltenden Bedingungen (den Policies) bzw. der Einwilligung für den

[70] vgl. [Grim03] zu den Möglichkeiten, mittels domainübergreifender Kooperation Cookies zur Kommunikation zwischen Anbietern zu nutzen und der damit verbundenen Komplexität, Cookies eindeutig einer bestimmten Kommunikationsbeziehung zuzuordnen.

[71] vgl. [Selk03]

[72] Dies setzt voraus, dass der Betroffene zu jedem Zeitpunkt dieselbe Entscheidung treffen würde, wenn dieselben Daten vom selben Verarbeiter erneut angefragt würden.

aktuell vorliegenden Fall. Die Kommunikation des Ergebnisses an den Verarbeiter erledigt wiederum das Gerät. Der Ablauf für eine Daten-Anfrage ist in Abbildung 6 dargestellt.

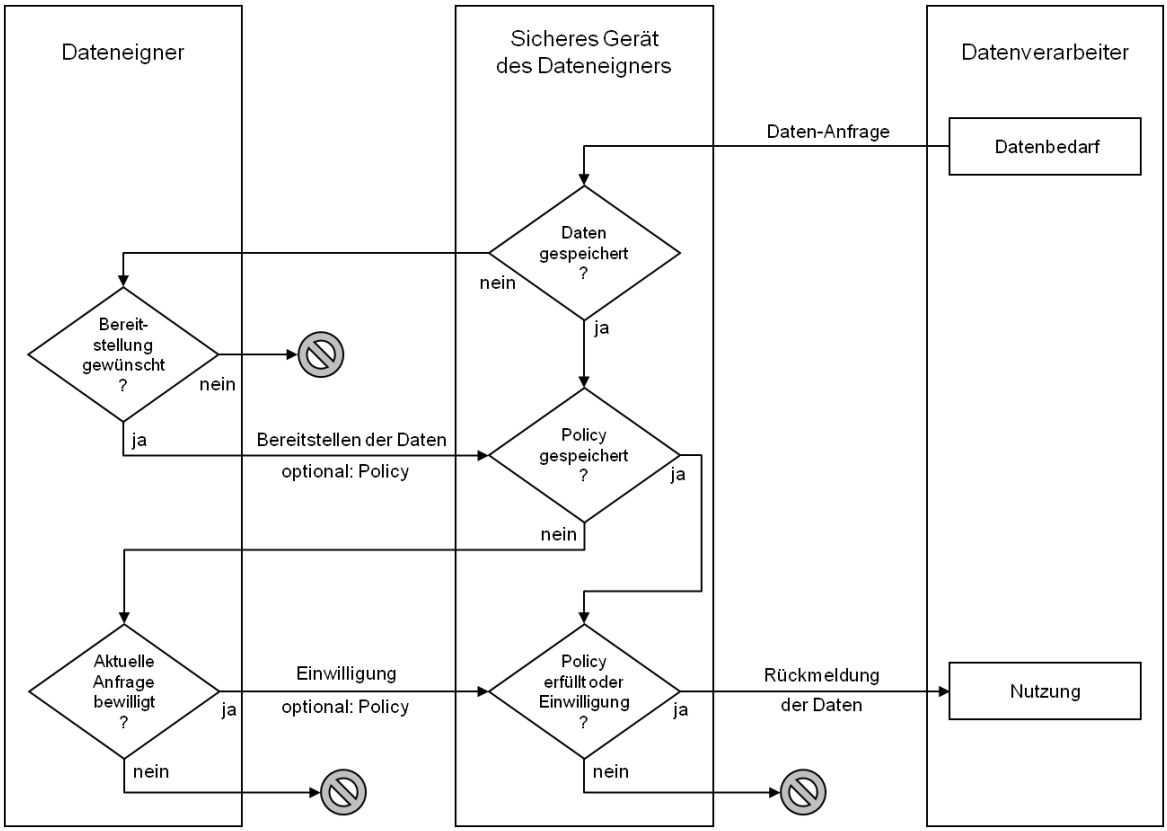


Abbildung 6: Prüfung einer Anfrage hinsichtlich Daten, Policy und Einwilligung

Das sichere Gerät müsste die folgenden grundlegenden Eigenschaften aufweisen:

- Eine **Anwender-Schnittstelle**, die dem Betroffenen vollständige Kontrolle über die Daten, Policies und die Verknüpfungen zwischen beiden erlaubt[73].
- Eine zweite Schnittstelle für die **Kommunikation mit dem Verarbeiter**. Über diese Schnittstelle werden generell keine Meta-Daten (Policies, Einwilligungen) nach außen gegeben. Der Anfragende erfährt nicht, ob ein verweigertes Datum überhaupt gespeichert ist, welche Policies hinterlegt sind und ob eine Entscheidung

[73] In einigen Fällen wäre das Verändern der hinterlegten Daten durch den Betroffenen für einen Vertragspartner nicht akzeptabel. Die Möglichkeit, beispielsweise das gespeicherte Meilenkonto im Rahmen eines Vielfliegerprogramms beliebig zu manipulieren, könnte von dem Anbieter sicher nicht hingenommen werden. Hier bietet sich an, die Daten vor Speicherung durch den Verarbeiter digital zu signieren. Durch Prüfung der Signatur kann beim erneuten Abruf der Informationen eine zwischenzeitliche Veränderung ausgeschlossen werden. Zu einem alternativen Vorschlag, konkret zu anonymen Kundenbindungssystemen, vgl. [EEOS05].

persönlich oder regelbasiert getroffen wurde. Der umgekehrte Weg, ein Import von Fremdpolicies, muss – für den Betroffenen überprüfbar – möglich sein.

- Bestmöglicher **Schutz der gespeicherten Daten** und Policies im Gerät gegen Ausspähen und Manipulation. Starke Verschlüsselung verhindert Zugriffe abseits der kontrollierten Schnittstellen. Die Schnittstellen selbst sind gegen Abhören, Verändern oder Unterbrechen der Kommunikation zu sichern.

- Zweckmäßig ist die Möglichkeit, den Inhalt des Gerätes auf einem **Backup-Medium** zu sichern, das dieselben Sicherheitsanforderungen erfüllt.

- Das Gerät soll sich unter ausschließlicher **physischer Kontrolle des Betroffenen** befinden. Es benötigt sichere Kommunikationsverbindungen, deren Verfügbarkeit und Qualität den Bedürfnissen der betroffenen Transaktionen Rechnung tragen.

- Zur Kommunikation mit den Verarbeitern, insbesondere zur Wiedererkennung durch diese, benötigt das Gerät Funktionen für ein **Identitätsmanagement**, die der folgende Abschnitt darstellt.

3.1.3 Kommunikation mit Verarbeitern

Sind alle personenbezogenen Daten ausschließlich beim Betroffenen gespeichert, stellt sich die Frage, wie grundsätzlich die Kommunikation vom Verarbeiter zum Betroffenen hergestellt werden kann. Es mag bisweilen erstrebenswert sein, dass nur der Betroffene die Möglichkeit besitzt, mit einem Verarbeiter in Kontakt zu treten. Vielfach ist dies jedoch nicht ausreichend, etwa im Sinne der Datenverarbeitung durch den Arbeitgeber oder für den automatischen Versand elektronischer Newsletter, die ein Betroffener abonniert hat. In diesen und weiteren Fällen ist es unvermeidbar, dass der Verarbeiter Kontaktinformationen zum Betroffenen bei sich speichert. Eine denkbare Lösung nutzt Pseudonyme:

Im sicheren Gerät ist ein Modul zum Identitätsmanagement[74] integriert, das für jede neue Geschäftsbeziehung Pseudonyme für den Betroffenen erzeugt[75]. Im einfachen Fall sind dies pseudonymisierte E-Mail-Adressen, an die ein Verarbeiter seine spezifischen Anfragen stellen kann. Dem Verarbeiter ist es erlaubt, bei sich Listen von Pseudonymen zu führen, mit deren Inhaber er Geschäftsbeziehungen unterhält. Um die zweckgerechte Verarbeitung der Pseudonyme zu ermöglichen, ist das Aufteilen auf verschiedene Listen durchaus sinnvoll. So kann es beispielsweise eine Liste mit Pseudonymen geben, deren Inhaber Angestellte eines Unternehmens sind, während eine andere Liste die Pseudonyme der Kunden desselben Unternehmens beinhaltet. Feinere Unterteilungen sind möglich,

[74] In der Definition nach [FeBe00], „Identitätsmanagement [...] soll einen Benutzer in die Lage versetzen, persönliche Merkmale nur gezielt und bewusst weiterzugeben.“, ist das beschriebene Gerät bereits ein Werkzeug des Identitätsmanagements.

[75] Zu den Arten von Pseudonymen vgl. [PfKö01], zur Handhabung, Aufdeckung und Verknüpfungsmöglichkeiten von Pseudonymen vgl. [BeKö01], zu Implementierung und Verwaltung derselben vgl. [FeBe00].

etwa innerhalb der Gruppe der Angestellten eine zusätzliche Teilliste der Pseudonyme behinderter Mitarbeiter, um entsprechende statistische Anforderungen von Behörden effizient beantworten zu können. Benötigt der Verarbeiter nun konkrete Daten zu den Pseudonymen einer ihm vorliegenden Liste, stellt er nacheinander entsprechende Anfragen an die hinterlegten Pseudonyme, im einfachen Fall also an die pseudonymen E-Mail-Adressen. Das sichere Gerät des Betroffenen isoliert das spezifische Pseudonym, verarbeitet die Anfrage, und antwortet entweder mit den angeforderten Daten oder einem negativen Bescheid ohne Nennung des Grundes. Eine Begründung könnte ungewollte Rückschlüsse zulassen. So kann es beispielsweise kompromittierend sein, wenn ein Betroffener die Herausgabe der Ergebnisse seines Gesundheits-Tests explizit verweigert. Lässt die Rückmeldung allerdings offen, ob überhaupt Ergebnisse eines Tests hinterlegt sind, ist der ungewollte Informationsgewinn beim Verarbeiter minimiert[76].

3.1.4 Strategien zur Datenweitergabe

Für die Lieferung der Policy-konformen Inhalte an den Anfragenden stehen verschiedene Optionen zur Verfügung. Lefevre et al. schildern mögliche Zugriffsstrategien (Limited Disclosure Modelle, LDM) im Zusammenhang mit den weiter unten erläuterten Hippokratischen Datenbanken. Eine analoge Anwendung auf die Datenbank des sicheren Geräts beim Betroffenen ist aufgrund des gemeinsamen Ziels erlaubt: „Specifically, we intercept an incoming query, and augment the query as necessary to reflect both the privacy policy and the donor's preferences.“[77]. In beiden Fällen ist eine Beschränkung von Abfragen auf der Ebene von Datensätzen nicht granular genug, so dass eine Betrachtung auf Feld-Ebene notwendig ist.

Im **Strict Cell-Level Enforcement** werden verbotene Werte – also Inhalte von Feldern, die weder den Anforderungen der Zweckbindung noch der Einwilligung genügen – in den Rückmeldungen zu Abfragen durch den Wert Null ersetzt. Das wird problematisch, falls als Rückmeldung ganze Tabellen erwartet werden und deren Schlüsselfelder von der Löschung betroffen sind. Daher werden zusätzlich im **Table Semantics Limited Disclosure Model** Datensätze komplett aus der Abfrage entfernt, wenn ihr Schlüssel einen verbotenen Wert enthält. Table Semantics LDM definiert eine Sicht der beteiligten Tabellen für jede Kombination aus Zweck und Empfänger. Diese Sichten dienen als Grundlage für die Abfragen. Im Gegensatz dazu findet im **Query Semantics Limited Disclosure Model** die Limitierung erst auf Ebene der Abfrage statt. Hier wird zunächst ein vollständiges Antwortset geliefert, aus dem dann die nicht erlaubten Werte oder Datensätze entfernt werden.

[76] Im Zusammenhang mit Limited Disclosure Modellen wird in [LAER+04] beispielhaft das SQL-Statement „SELECT * FROM PATIENTS WHERE DISEASE = Hepatitis“ angeführt. Die Tatsache, dass ein bestimmter Name zurückgeliefert wird, lässt auf die gewünschte Information schließen, auch wenn die tatsächliche Diagnose nicht ausgegeben wird.

[77] [LAER+04]

Dementsprechend bestehen zwei Optionen zur Behandlung eingehender Abfragen. Entweder werden die Abfrage-Statements automatisch umgeschrieben, wobei Policies als zusätzliche Restriktionen aufgenommen werden. Oder die Abfragen bleiben unverändert, werden aber nicht über die eigentlichen Tabellen ausgeführt, sondern über Sichten, die für jede Kombination aus Zweck und Empfänger pro Tabelle angelegt wurden. Diese Anlage kann entweder zur Laufzeit erfolgen oder bereits vor der operativen Nutzung geschehen.

3.1.5 Analyse des Datenschutzniveaus

Zur Einordnung des Datenschutzniveaus werden die von Gola und Japsers[78] aus dem Bundesdatenschutzgesetz (BDSG) abgeleiteten 7 „Säulen“ des Datenschutzes verwendet: Zulässigkeit, Zweckbindung und Datenvermeidung, Transparenz, Korrekturrechte, Datensicherung, Kontrolle und Sanktionen.

- **Zulässigkeit**

  Das Bundesdatenschutzgesetz fordert grundsätzlich das Vorliegen einer besonderen Rechtfertigung für die Erhebung, Verarbeitung und Nutzung personenbezogener Daten. Diese kann entweder durch eine Einwilligung des Betroffenen oder durch Rechtsvorschrift (Zulässigkeitstatbestände des BDSG oder Gestattung bzw. Verpflichtung durch Rechtsvorschriften außerhalb des BDSG) gegeben sein.

  Roßnagel, Pfitzmann und Garstka formulieren vier Bedingungen für das Vorliegen einer gültigen Einwilligung[79]: (1) Eine **ausdrückliche Erklärung** des Betroffenen liegt automatisch vor, wenn er eine konkret vom Verarbeiter vorliegende Anfrage beantwortet. Die angeforderten personenbezogenen Daten wird er in dieser Antwort nur dann übermitteln, wenn er sich über deren Nutzung (2) **vollständig unterrichtet** fühlt. Eine (3) **unzulässige Kopplung**, das heißt das Erheben von zusätzlichen Daten, die zur Erfüllung der vertraglichen Vereinbarung nicht notwendig sind, wird für den Datenverarbeiter unattraktiv, da ihm im vorliegenden Modell die Speicherungsmöglichkeit fehlt. Oft steht die (4) **Freiwilligkeit** der Einwilligung in Frage, weil ein Machtgefälle gegenüber dem Verarbeiter dem Betroffenen keine echte Wahl lässt[80]. Eine umfassende Lösung dieses Problems ist bislang nicht bekannt. Wir gehen jedoch davon aus, dass die Verweigerung der Einwilligung als personenbezogenes Datum ebenfalls nicht beim Verarbeiter gespeichert werden kann. So ist der Betroffene vor einer späteren Benachteiligung aufgrund dieser Weigerung im bestmöglichen Maße geschützt.

  Ist im Gegensatz zur Einwilligung die Datenerhebung durch Rechtsvorschriften begründet, liegt die Macht zur Herausgabe ebenfalls beim Betroffenen. Er kann die

[78] [GoJa01]
[79] vgl. [RoPG01] S.91ff
[80] etwa im Arbeitsverhältnis, vgl. [Däub02] und allgemein [GoSK05] §4a Rn6.

Rechtmäßigkeit der Ansprüche prüfen oder von unabhängiger Stelle prüfen lassen. Alternativ könnte der Verarbeiter die Rechtmäßigkeit seiner Erhebungsansprüche bereits vorab durch eine unabhängige Stelle zertifizieren lassen und dieses Zertifikat vorzeigen. Verweigert der Betroffene dennoch die Herausgabe der Daten, muss der Verarbeiter den Rechtsweg bemühen.

- **Zweckbindung / Datenvermeidung**

  Die Zweckbindung von Datenverarbeitung tritt als eigenständiger Grundsatz in den Hintergrund, wenn alle Daten beim Betroffenen exklusiv gespeichert sind. Denn sofern der Betroffene seine Daten herausgibt und kein durch Rechtsvorschrift legitimierter Zweck dafür vorliegt, ist seine Einwilligung zu unterstellen. Ein Verarbeiter könnte diese zwar durch Täuschung erschlichen haben oder im Moment der Datenübermittlung den Nutzungszweck ändern. Aber auch hier könnte eine unabhängige Zertifizierung für Stärkung der Vertrauensbasis sorgen.

- **Transparenz**

  Es ist bereits im Design des Verfahrens festgelegt, dass Datenverarbeitung nur nach unmittelbarer Einwilligung des Betroffenen stattfinden kann. Dass diese Einwilligung nur unter ausreichender Informiertheit erfolgen wird, wurde gezeigt. Da jede zusätzliche Verarbeitung von erhobenen Daten mangels Zwischen-speicherung unmöglich ist, entfällt die Benachrichtigung bei derselben. Aus den gleichen Gründen sind spätere Ausweitungen oder Umwidmungen des Verwendungszwecks sowie die Weitergabe der Daten an Dritte ausgeschlossen. Für Benachrichtigungen besteht also kein Grund, die Information des Betroffenen über die Nutzung seiner Daten ist maximal.

- **Korrekturrechte**

  Eine Stärke liegt offenkundig in der Wahrung der Rechte des Betroffenen auf Berichtigung, Sperrung und Löschung seiner Daten. Diese Aktivitäten werden mit unmittelbarer Wirkung am Speicherort der Daten durchgeführt. Dieser liegt beim Betroffenen, der die volle Kontrolle über die Qualität und Verfügbarkeit seiner Daten innehat. Auch ein Widerspruch gegen die Nutzung der personenbezogenen Daten oder deren Weitergabe zu Werbezwecken wird zu den Korrekturrechten gezählt. Widerspruch kann sinnvoll eingelegt werden, wenn zuvor Kontrolle abgegeben wurde, etwa aufgrund einer zeitlich unbefristeten Einwilligung. Dies ist im Modell ausgeschlossen, daher kommt dem Widerspruch als eigenständige Aktivität keine Bedeutung mehr zu. Seine datenschützerische Intention ist implizit im Modell abgedeckt.

- **Datensicherung**

  Die Qualität der Datensicherung hängt von der technischen Ausstattung und den Kenntnissen des Betroffenen ab. Diese dürften regelmäßig nicht mit den Kapazitäten großer Unternehmen Schritt halten können. Ein interessierter und informierter Computernutzer wird sich der Herausforderung stellen.

Anwendungsorientierte Betroffene und Nutzer von Geräten, die manuelle Konfiguration nur begrenzt zulassen, Mobiltelefone, Embedded Systems oder PDAs mit proprietären Betriebssystemen, müssen sich mehr oder weniger auf bestehende Produktkonfigurationen verlassen. In jedem Fall sind Vorkehrungen für verschiedene Aspekte der Datensicherung zu treffen[81].

Im Modell sind die personenbezogenen Daten eines Menschen an einer Stelle gespeichert, deren Ausgestaltung dem Verantwortungsbewusstsein des Einzelnen überlassen ist. Einheitliche Datensicherungsstandards könnten durch die Definition eines ausreichend sicheren Geräteprofils erreicht werden. Die Auslagerung der Datenspeicherung an einen Dienstleister, der eine sichere Infrastruktur bereithält, könnte eine Alternative sein. Allerdings opfert sie Vorteile, die das Modell einer exklusiven Speicherung beim Betroffenen mit sich brächte, beispielsweise die vollständige Transparenz oder das Ausüben der Korrekturrechte ohne Zeitverzug.

- **Kontrolle und Sanktionen**

  Interne Kontrolle durch betriebliche oder behördliche Datenschutzbeauftragte, externe Kontrolle durch die Aufsichtsbehörden, sowie durch diese angestoßene Sanktionen im Fall von Verstößen vervollständigen die sieben Säulen des Datenschutzes. Diese Aspekte rechtlich-organisatorischer Art würden bei tatsächlicher Anwendung des beschriebenen Modells einer Neugestaltung bedürfen, die hier aufgrund der theoretischen Natur des Verfahrens beiseitegelassen wird.

3.1.6 Einsatztauglichkeit für hoheitlichen Datenzugriff

Mit dem vorgestellten Modell der ausschließlichen Speicherung beim Betroffenen wäre zweifellos ein hohes Datenschutzniveau zu erzielen. Der Betroffene hätte zu jeder Zeit die volle Kontrolle über seine Daten, kein Verarbeiter würde erhaltene Daten über den Moment der konkret zugelassenen Nutzung hinaus speichern, der Betroffene könnte jederzeit die Zugriffs-Policies ändern oder seine Einwilligung in konkreten Einzelfällen entziehen. Einem praktischen Einsatz stehen jedoch fundamentale Kritikpunkte entgegen, die nicht vollständig ausgeräumt werden können:

Hoheitlich tätige Organisationen benötigen Datenzugriff, der zum Teil unbeobachtet, aber in jedem Fall ohne Eingriffsmöglichkeiten des Betroffenen, stattfindet. Zu diesen Organisationen gehören Polizei, Geheimdienste, Sozialbehörden und der Zoll[82]. Zur Abwehr terroristischer Bedrohungen, zur Verfolgung und Verhinderung von Betrug und

[81] Münch erläutert ausführlich die „8 Gebote“ der Datensicherung gemäß Anhang zu §9 BDSG, vgl. [Münc05], [Hauf11]: Zutrittskontrolle, Zugangskontrolle, Zugriffskontrolle, Weitergabekontrolle, Eingabekontrolle, Auftragskontrolle, Verfügbarkeitskontrolle, Trennungsgebot.

[82] Bizer sieht im Verhältnis zu diesen Behörden in der Datenverarbeitung, die exklusiv beim Betroffenen stattfindet einen Verstoß unter anderem gegen das Rechtsstaatsgebot, die Pflicht der nachvollziehbaren Dokumentation des Verwaltungshandelns und den Nachweis der Rechtmäßigkeit in Verwaltungsvorgängen. Sie berge außerdem erhebliche Beweisrisiken in gerichtlichen Streitverfahren, vgl. [Bize02].

Verbrechen sind diese Institutionen mit besonderen Rechten ausgestattet. Hier scheint eine vollständige Autonomie des Betroffenen nicht durchsetzbar. Da eine Manipulation am Datengerät nicht vollständig ausgeschlossen werden kann oder vielleicht sogar explizit vorgesehen sein könnte, scheint ein Sonderzugriffsrecht der genannten Institutionen auf den Datenspeicher nicht ausreichend. Es ist zudem fraglich, ob eine Verfügbarkeit des Datengeräts unter vertretbaren Kosten in vollem Umfang sichergestellt werden kann. Die Möglichkeit einer vollständig unbeobachtbaren Datenabfrage durch bestimmte Stellen müsste durch Backdoor-Funktionen ermöglicht werden[83]. Ein Verdächtiger könnte bereits dann die Flucht ergreifen, wenn er bemerkt, dass eine fremde Datenverbindung zu seinem sicheren Gerät hergestellt wurde. Das Entdecken solcher Anfragen könnte durch das regelmäßige Generieren von Dummy-Datenpaketen verschleiert werden, die zuverlässig wie echte Datenpakete aussehen. Zusätzlich müssten die Datenbankabfragen durch befugte Stellen so gestaltet sein, dass die Datenbank selbst nicht nachweisen kann, welcher Datensatz abgefragt wurde[84].

Datenspeicher bei den hoheitlich tätigen Stellen scheinen also im Widerspruch zur eingangs formulierten Prämisse der exklusiven Speicherung beim Betroffenen unausweichlich. Natürlich ist deren Nutzung rigider Kontrolle zu unterstellen, insbesondere sind die Modalitäten von Weitergabe und Data Mining streng zu regulieren.

3.1.7 Anwendungsbezogene Problemstellungen

Hohes Datenschutzniveau, das im Alltag funktioniert, kann in Ausnahmesituationen für den Betroffenen unerwünschte Auswirkungen haben, zum Beispiel im medizinischen Bereich. Sind alle personenbezogenen Daten nur beim Betroffenen gespeichert, haben Notärzte keinen Anhaltspunkt, wo sie einen bedrohlich Erkrankten finden, wenn es diesem nicht mehr möglich ist, seine Adresse im Notfall weiterzugeben. Organspenden finden keinen Empfänger, da es keine Datei der geeigneten Patienten gibt. Die Verwendung von Pseudonymen könnte hier hilfreich sein (Rollen- und/oder „Geschäftsbeziehungs"-Pseudonyme, etwa für Ärzte, Apotheken und Giftambulanz im gesundheitlichen Notfall). Um einer Person im Notfall helfen zu können ist für einen alarmierten Helfer zunächst keine exakte Personenzuordnung notwendig, solange unter dem verwendeten Pseudonym die Person aufgefunden werden kann.[85]

Eine Reihe von Verarbeitungen legt die Aggregation von Daten mehrerer Personen zugrunde. Diese können etwa wissenschaftlichen Zwecken entspringen oder der

[83] Brands hat einige Beispiele dafür zusammengetragen, dass diesem Wunsch verschiedentlich nachgekommen wird, vgl. [Bran00].

[84] Vorschläge zu einem solchen Algorithmus liefern Asonov und Freytag [AsFr02].

[85] Die elektronische Gesundheitskarte, deren Sicherheitskonzept vorsieht, dass ein Patient den Zugriff von medizinischem Personal auf seine Daten durch seine persönliche PIN autorisiert, reserviert für Notfalldaten einen Speicherbereich, für den die PIN nicht notwendig ist. Auf die dort gespeicherten Daten sollen behandelnde Personen mit ihrem Heilberufsausweis zugreifen können. Vgl. http://www.bmg.bund.de/krankenversicherung/elektronische-gesundheitskarte/glossar-elektronische-gesundheitskarte.html#c14550 (Zugriff: 19.05.2012).

Marktforschung dienen. Auch wenn es für diese Verarbeitungen zumeist ausreichend ist, Daten anonym und nur aggregiert zu speichern, sind zunächst die Einzelwerte zu erfassen, bevor sie verdichtet werden können. Verfahren zur „Secure Multi-Party Computation“[86] können in einigen Konstellation helfen, aber da die Erfassung möglicherweise über einen längeren Zeitraum hinweg erfolgt, ist eine Speicherung der Einzelwerte in anderen Fällen unumgänglich.

Die Manipulation eines Betroffenen am Datengerät ist nicht auszuschließen, da es sich exklusiv in seiner Kontrolle befindet. Auch der Nachweis einer Manipulation kann zunächst nicht erbracht werden, wenn keinerlei Daten das Gerät ohne Einwilligung des Betroffenen verlassen. Ein Datenverarbeiter, der mit Einwilligung des Betroffenen also Daten auf dessen Gerät speichert, etwa den Meilenstand eines Programms zur Kundenbindung, kann sich ohne Kontrollmechanismen nicht sicher sein, beim nächsten Auslesevorgang dieselben Daten unverändert vorzufinden[87]. Er kann sich jedoch schützen, indem er die Inhalte vor dem Speichern signiert und damit vor Verfälschung schützt. Der Betroffenen wird jedoch regelmäßig nicht zulassen wollen, dass andere ihre Daten verschlüsselt auf dem Gerät speichern. Denn dies wäre eine Art „Cookie-Ersatz“ außerhalb der Kontrolle des Betroffenen.

Problematisch ist das Modell einer exklusiven Speicherung beim Betroffenen weiterhin, wenn aus einer Menge von Personen aufgrund der von ihnen gespeicherten personenbezogenen Daten eine Auswahl zu treffen ist. Dies kann die Registrierung für ein Gewinnspiel ebenso sein, wie der Eintrag in einer Bewerberdatenbank, bei einem potentiellen Arbeitgeber oder in Jobbörsen. Diese Systeme beruhen darauf, eine Menge von Personen anhand merkmalsabhängiger Kriterien zu gruppieren („Nimmt am Gewinnspiel teil“ oder „Hat mehr als drei Jahre Berufserfahrung in Netzwerktechnik“) und basierend auf den personenbezogenen Daten eine Auswahl zu treffen (Auslosung durch Ziehen eines Namens oder Abgleich mit einem Anforderungsprofil). Im Versuch, Mehrfachteilnahmen oder -bewerbungen zu verhindern, könnte hierbei in vielen Fällen Pseudonymisierung durch entsprechende Nutzungsbedingungen ausgeschlossen sein. Lösungen für dieses Problem zu finden, muss künftiger Forschungsarbeit überlassen bleiben.

[86] vgl. [DuAt01], [OnLZ09]. Ganz grundlegend werden beispielsweise Summen sicher gebildet, indem ein Teilnehmer eine Zufallszahl erzeugt, und sie geheimhält. Er addiert den von ihm zu liefernden Wert zu der Zufallszahl und gibt die Summe an den nächsten Teilnehmer. Reihum addieren alle Teilnehmer ihre Werte. Zuletzt erhält der erste Teilnehmer die Gesamtsumme, subtrahiert seine Zufallszahl und stellt den sicher berechneten Wert der Allgemeinheit zur Verfügung. Vgl. zu dieser „Secure Sum“ [Dong09].

[87] Roßnagel/Pfitzmann/Garstka stellen heraus, dass personenbezogene Daten nicht nur Daten der betroffenen Person seien, sondern ebenso der Stelle, die die Daten erhoben oder verarbeitet hat: „So sind Daten über eine medizinische Behandlung zugleich auch Daten über die Leistung des Arztes, die dieser benötigt, um seinen Leistungsanspruch zu begründen und abzurechnen, um seine ärztliche Dokumentationspflicht zu erfüllen und im Streitfall eine ordnungsgemäße Behandlung nachweisen zu können.“ [RoPG01]

3.1.8 Zusammenfassung der Untersuchung

Die Speicherung beim Betroffenen als Modell möchte man zunächst mit Bizer beurteilen: „Die Smart Card würde [...] als lokaler Datenspeicher in der Hand des Bürgers das Prinzip der Datenvermeidung technisch ermöglichen und gleichzeitig die informationelle Selbstbestimmung des Bürgers als Bestimmungsbefugnis über die eigenen Daten in idealer Weise verwirklichen.“[88] Ebenso muss man sich aber anschließen, wenn er nach Betrachtung der Hürden fortfährt: „Vor dem Hintergrund derartiger Probleme verliert die Vision der Smart Card als Datenspeicher für Verwaltungsunterlagen ‚in der Hand‘ des Bürgers einen Großteil ihrer datenschutzrechtlichen Faszination.“[89] So zeigt auch die oben durchgeführte Analyse, dass besonders die normativen Rahmenbedingungen wie hoheitliche Tätigkeiten und Vertragsrecht die Hürden vor einem stringenten Einsatz des Modells darstellen. Illusion bleibt freilich vor allem die Annahme, dass eine Speicherung personenbezogener Daten im Kontrollbereich der Verarbeiter unterbunden werden könnte.

Das theoretisch hohe Datenschutz-Niveau, das dem Konzept innewohnt, kann man anhand der in Abschnitt 2.3 formulierten Wirkungsdimensionen überprüfen: Der Formulierungsort der Policies liegt beim Betroffenen, er hat die maximale Kontrolle über die Art der Formulierung, den Inhalt und auch den Lebenszyklus. Er kann zu jedem beliebigen Zeitpunkt die Policies ändern und die Änderung wirksam werden lassen, sofern dem keine juristischen Hürden im Wege stehen. Die Bindung der Policies an die Daten ist in diesem theoretischen Modell, das einem Verarbeiter keine Speicherungsmöglichkeiten zugesteht, technisch nicht umgehbar, also maximal. Im Datenlebenszyklus werden die Phasen bis „Verteilen“ umfangreich abgedeckt, während die folgenden Phasen, die sich auf die beim Verarbeiter gespeicherten Daten beziehen, hier nicht existieren. Die Durchsetzung der Policies erfolgt beim Betroffenen und präventiv vor Übermittlung der Daten. In diesem Sinne ist das Modell auf ein Minimum an notwendigem Vertrauen gegenüber Verarbeitern oder Dritten ausgerichtet. In entgegengesetzter Richtung müssen die Verarbeiter dem Betroffenen ohne Sicherungsmöglichkeiten ihr Vertrauen entgegenbringen. Diese in vielen Fällen nicht zu akzeptierende Situation und das praktische Unvermögen, das Speichern von Daten durch Verarbeiter zu unterbinden, verweisen den Ansatz denn auch in den Bereich der Theorie.

[88] [Bize02]
[89] Ebenda

Speicherung beim Betroffenen

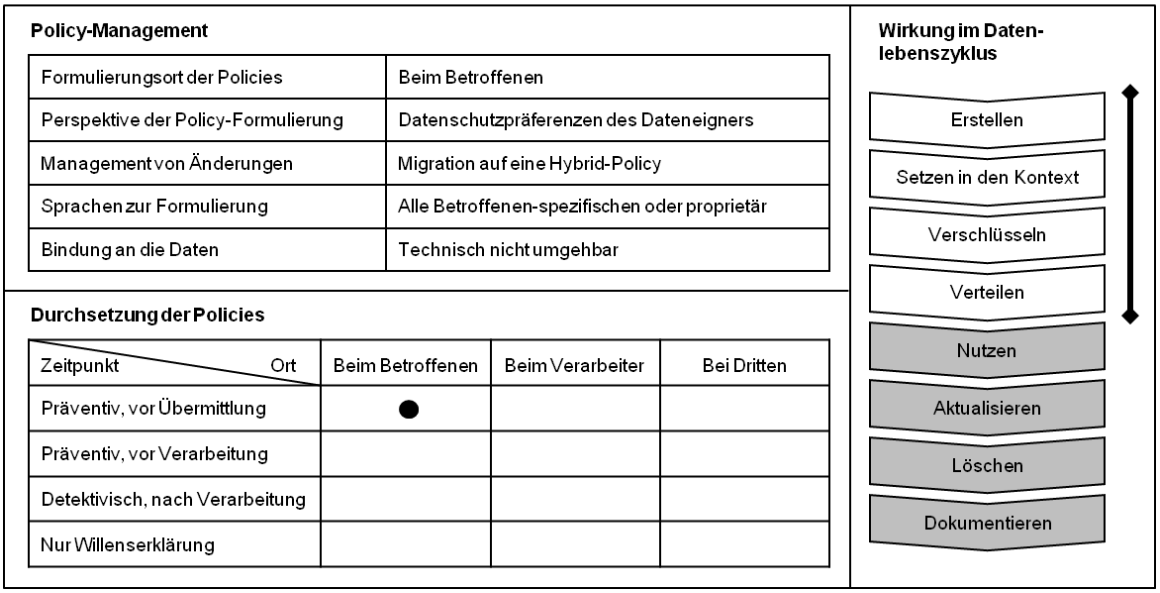


Abbildung 7: Steckbrief Speicherung beim Betroffenen

3.2 Access Control

3.2.1 Einfache Access Control

Access Control (Zugriffskontrolle) ist eine Sicherheitsfunktion, die Ressourcen gegen unberechtigten Zugriff schützt. Die Ressourcen können dabei technischer Natur sein (d.h. die Nutzung von Hard- und Software), aber ebenso Funktionen eines Systems oder dort gespeicherte Daten.[90] Die Unterscheidung zwischen berechtigter und unberechtigter Nutzung treffen Access Control Regeln (Policies), umgesetzt in **System-Berechtigungen**, die basierend auf den Schutzbedürfnissen modelliert und durch Administratoren den Nutzern zugeordnet werden. Systeme, die Access Control implementieren, beinhalten zu diesem Zweck mindestens drei Komponenten: Einen **Policy-Editor**, einen Mechanismus zur **Policy-Durchsetzung** und eine **Entscheidungsfunktion**. Der Policy-Editor ermöglicht das Erstellen der maschinenlesbaren Zugriffsregeln auf Basis der verfügbaren Ressourcen, der potentiellen Nutzer und den möglichen Aktionen. Der Durchsetzungs-mechanismus fängt alle Zugriffsanforderungen ab und befragt die Entscheidungsfunktion darüber, ob gemäß der Policies der Zugriff zu gestatten oder zu verweigern ist. Nur bei Übereinstimmung der Anforderung mit den geltenden Policies wird der Zugriff durch den Durchsetzungsmechanismus gewährt. Dabei ist die wichtigste Eigenschaft des Durchsetzungsmechanismus, dass er potentiell alle eingehenden Zugriffsanforderungen

[90] vgl. [SHPR+06], Zugriffskontrollen in komplexen Systemen können nur wirksam sein, wenn sie die zentralen Komponenten integrativ betrachten, also unter einen gemeinsamen Kontrollschirm nehmen.

abweisen kann, das heißt, dass keine Möglichkeiten bestehen, den Mechanismus zu umgehen.[91]

Das Gewähren von Zugriffsberechtigungen kann einerseits vom Eigentümer des zu schützenden Objekts ausgehen (Discretionary Access Control – DAC) oder vom System vorgegeben werden (Mandatory Access Control – MAC).[92]

Seit Rechner-Ressourcen mehr als einer Person zur Verfügung stehen, ist Access Control von Bedeutung. Ging es zunächst darum, kostbare Rechenzeit und Speicherkapazitäten zu sichern, stehen mit deren mittlerweile umfassender Verfügbarkeit die Schutzbedürfnisse von Geschäftsgeheimnissen, personenbezogenen Daten und öffentlicher Präsenz im Vordergrund. Dementsprechend waren auch die hergebrachten Modellierungsmöglichkeiten für die Access Control Policies nicht mehr ausreichend. Die klassische Abbildung in Form einer Matrix mit den Nutzern als Zeilen, den Ressourcen in den Spalten und den erlaubten Aktivitäten als Einträge in den Zellen[93] stößt in komplexen Systemen schnell an die Grenzen der Wartbarkeit und erlaubt darüber hinaus nicht die Modellierung von Abhängigkeiten, Bedingungen und dynamischen Konzepten[94], wie sie insbesondere auch im Dienste des Datenschutzes benötigt werden. Zweckbindung und Einwilligung bleiben außen vor.

So wie die klassische Access Control im Hinblick auf weitgehenden Datenschutz Lücken aufweist[95], können auch die zahlreichen vorgestellten Erweiterungen dieselben nicht vollständig schließen. Die wichtigsten Entwicklungen seien dennoch im Folgenden kurz dargestellt[96].

3.2.2 Role based Access Control

Role based Access Control (RBAC) knüpft die Berechtigungen zur Verarbeitung geschützter Objekte an **Rollen**, die eine Menge sachlogisch miteinander verbundener Aufgaben darstellen. Das Modell legt vorab fest, welche Verarbeiter welche Rollen wahrnehmen und damit auch, für welche Aufgaben sie berechtigt sind[97]. Ein Verarbeiter kann Inhaber mehrerer Rollen sein und alle ihm zugeordneten Rollen sind gleichzeitig aktiv. Der Verarbeiter besitzt immer die in allen ihm zugewiesenen Rollen modellierten Zugriffsberechtigungen. Systeme mit rollenbasierter Zugriffskontrolle konzentrieren ihre Zugriffsbeschränkungen üblicherweise auf die im Rahmen einer Prozessanalyse identifizierten Aufgabenbereiche der Anwender. Die implementierten Rollen können,

[91] vgl. [Tilb05] „Access Control“
[92] vgl. [KoMP01]
[93] „Access Matrix Models“, zumeist implementiert als „Access Control Lists“ (ACL).
[94] vgl. [KaPF01], die zeigen, wie klassische Access Control die unterschiedlichen Aufgaben, die ein Nutzer als Teilnehmer eines Workflows einnimmt, nicht abdeckt.
[95] „It is necessary to build „privacy extensions“ of traditional access control systems that can author and enforce privacy policies.” [CTBC05]
[96] Ausführlichere Behandlung etwa in [Bena05].
[97] vgl. [Ecke04]

müssen aber keineswegs im Zusammenhang mit den Datenschutzregelungen stehen, je nachdem, ob diese beim Design des Rollenkonzepts berücksichtigt wurden. RBAC ist das Berechtigungssystem der Wahl für die Mehrzahl der heute im Einsatz befindlichen Enterprise Resource Planning-Systeme (einschließlich der Personalverwaltungssysteme) wie die Business-Suiten von SAP oder Oracle[98]. Die Zusammenfassung von Aufgaben in Rollen ermöglicht eine relativ einfache Identifikation und Durchsetzung von gegenseitigen Ausschlüssen, so dass im Sinne des 4-Augen-Prinzips kritische Kombinationen von Aufgaben nicht gemeinsam an einen Verarbeiter vergeben werden. Soll etwa in der Personalverwaltung vermieden werden, dass im Bewerbungsmanagement die Behinderung von Mitarbeitern ihre Chancen auf eine neue Stelle schmälert, so wird man die Rollen zum Bewerbungsmanagement und zur Anfertigung der Meldung über die Beschäftigung schwer behinderter Menschen an die Agentur für Arbeit unterschiedlichen Mitarbeitern zuordnen.

3.2.3 Authorization based Access Control

Alan Karp kritisiert die dargestellten Ansätze, Zugriffsrechte oder Rollen an den vorgewiesenen Identitäten von Verarbeitern festzumachen[99], die wiederum im Policy-System des Systems nachgeschlagen werden müssen, um ihre Rechte zu identifizieren. Insbesondere weist er darauf hin, dass der Diebstahl einer Identität alle hierfür hinterlegten Policies kompromittieren würde. Seine Lösung besteht in der **direkten Zuweisung** von Berechtigungen (Authorizations) an Verarbeiter, die sie analog ihrer Identitäts-Credentials direkt bei sich tragen, beim Versuch des Zugriffs auf geschützte Daten vorzeigen und sogar an für sie vertrauenswürdige Verarbeiter weitergeben können. Natürlich bleibt das Problem bestehen, Identitäten und Policies zuverlässig zusammen zu bringen, nur wird es im Lebenszyklus der Berechtigungen verlagert.

3.2.4 Task based Access Control

Task based Access Control (TBAC) unterstützt die dynamische Aktivierung von Zugriffsberechtigungen im Kontext der gerade in Bearbeitung befindlichen Aktivitäten. So besteht ein Zugriff anders als im herkömmlichen RBAC nur für den **Zeitraum**, der für eine spezifische Aktivität benötigt wird[100]. Zugriffsrechte erhalten also einen eigenen Lebenszyklus, der sowohl von externen Ereignissen wie auch von den Beziehungen zu anderen Zugriffsrechten abhängig sein kann. So kann etwa in einem Personalverwaltungssystem dem Sachbearbeiter nur für den Zeitraum der Abrechnungsvorbereitung Zugriff auf die Stammdaten der Mitarbeiter gewährt und danach wieder entzogen werden, ohne dass manuelle Rechtezuweisung und

[98] vgl. zu SAP [Horn00], Oracle [Abel06], SAP HCM [EsJu08], SAP ERP [LeSt09].
[99] vgl. [Karp06]. Diese Ansätze nennt er „Identity-based Access Control“ (IBAC), in Abgrenzung zu seinem Vorschlag der “Authorization-based Access Control” (ABAC).
[100] vgl. [ThSa97], [KüWe05]

-entzug durch Administratoren stattfinden müssten. Ein anderes Beispiel sind die erweiterten Zugriffsrechte für manche Nutzer, die nur im zeitlich begrenzten Ablauf einer Prüfung des internen Kontrollsystems gültig sind, oder die für die Dauer der Behebung eines akuten Systemfehlers erforderlich sind. Schließlich kann auch das Vorliegen einer Einwilligung zur Verarbeitung die dynamische aufgabenbezogene Erteilung von Zugriffsrechten auslösen. TBAC erfüllt durch die Möglichkeiten, Zugriffsrechte abhängig von Einwilligungen und Verarbeitungszwecken zu erteilen, grundlegende Anforderungen an einen datenschutzfreundlichen Einsatz. Die dazu benötigten granularen und applikations- wie feldbezogenen Policies können praktisch jedoch nur durch eine zentrale Instanz innerhalb einer Organisation erstellt werden. Somit eignet sich das Konzept nicht für unternehmensübergreifende Kommunikation[101].

3.2.5 Concept-level Access Control

Visionär über die Beschränkungen der Zugriffskontrolle auf einzelne Daten-Elemente hinaus geht die **semantische** oder „Concept-level" Access Control[102]. Hier sind die zu schützenden Inhalte mit entsprechenden Ontologien verbunden, die ganze Konzepte, das heißt vollständige Begriffswelten im Kontext des Schutzgegenstandes sichern. Vorausgesetzt, die geeigneten Ontologien[103] wären allgemein verfügbar, müsste man anstelle einzelner benannter Datenfelder nur das Konzept „Personenbezogene Daten" in Zugriffskontroll-Policies aufnehmen, um alle relevanten Daten zu schützen.

3.2.6 Eigner-kontrollierte Access Control

Weitere Konzepte wie Owner-Retained Access Control (ORAC)[104] und Originator-controlled Access Control (ORCON)[105] schlagen den Weg in die Richtung eines **digitalen Rechtemanagements** ein, indem sie der Access Control Elemente zum Schutz von Daten nach ihrer Weitergabe und Nutzung durch Verarbeiter hinzufügen. Verarbeiter, die möglicherweise im zeitlichen Verlauf als neue Eigner eines Datums auftreten, können die ursprüngliche Policy nicht ändern. Über die Policy hinausgehende Verarbeitungen müssen vom ursprünglichen Eigner explizit genehmigt werden. Ohne umfassende Kontrolle über die IT-Umgebungen aller potentiellen Verarbeiter sind diese Konzepte jedoch nicht umfänglich umsetzbar.

[101] vgl. [KaSW02]
[102] vgl. [QiAt04]
[103] Zur allgemeinen Einführung in die Techniken der semantischen Modellierung siehe [Hjel01], [Powe03].
[104] vgl. [CoMN90]
[105] vgl. [Sant09], ebenfalls zu den möglichen Einsatzszenarien.

3.2.7 Usage Control

Die verschiedenen Ausprägungen und Erweiterungen der Access Control lassen sich in einen gemeinsamen Rahmen einfügen, der eine einheitliche Beschreibung und Modellierung erlaubt. Park und Sandhu schlagen dazu das Modell **UCON$_{ABC}$** vor[106] (Usage Control). ABC steht dabei für die Prädikate nach denen sich Nutzungsentscheidungen modellieren lassen: (A)uthorizations (Berechtigungen), o(B)ligations (Verpflichtungen) und (C)onditions (Bedingungen). Die Kernelemente von UCON$_{ABC}$ zeigt Abbildung 8.

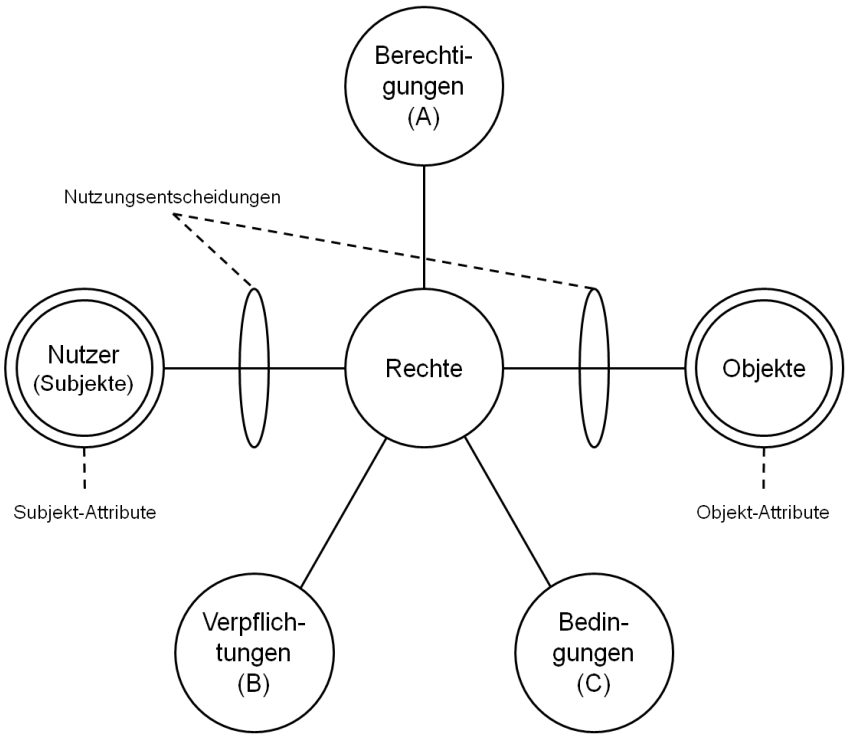


Abbildung 8: Elemente des UCON$_{ABC}$-Modells[107]

Ein Nutzer (Subjekt) repräsentiert in dem Modell ein Individuum, das Rechte an Objekten hält oder ausübt. Je nach Einsatzgebiet kann das Subjekt eindeutig identifizierbar sein oder anonym auftreten. Es verfügt über Attribute, die es beschreiben. Die Attribute können veränderbar oder unveränderbar sein. Ein unveränderliches Attribut könnte die Identität des Subjekts sein, ein veränderliches Attribut dem gegenüber seine aktuellen Credentials. UCON kann über veränderliche Attribute auch Modelle des Digital Rights Management (DRM) abbilden. Hier wäre ein Attribut etwa die verbleibende Anzahl von erlaubten Abrufen digitaler Inhalte.

Objekte sind Entitäten, an denen Subjekte Rechte halten, um auf sie zuzugreifen oder sie zu nutzen. Auch Objekte verfügen über Attribute, ebenfalls veränderbar oder

[106] vgl. [PaSa02], [SaPa03]
[107] nach [PaSa04]

unveränderlich. In den Attributen wird beispielsweise festgehalten, zu welchen Kategorien die Objekte gehören oder wer ihr Eigner ist.

Rechte sind die Mengen an Nutzungsfunktionen, die Subjekten den Zugriff auf Objekte gewähren. Die bekanntesten Rechte sind die zur Anlage, zur Ansicht, zur Änderung und zum Löschen eines Objekts, in DRM-Szenarien wären es die Ausgabe, die Weitergabe oder die Nutzung in anderen Inhalten. Darüber, ob ein Subjekt die jeweils benötigten Rechte besitzt, wird in $UCON_{ABC}$ zur Laufzeit der Anfrage entschieden, indem die Ausprägungen der Subjekt- und Objekt-Attribute, die Berechtigungen, die Verpflichtungen und Bedingungen ausgewertet werden. Es besteht also keine festgelegte Matrix von Rechten. Sofern veränderliche Attribute in die Nutzungsentscheidungen einfließen sollen, wäre eine vorgelagerte Festlegung auch gar nicht möglich.

Berechtigungen sind Regeln, die für die Nutzungsentscheidungen ausgewertet werden und bestimmen, ob ein Subjekt ein bestimmtes Recht auf ein Objekt ausüben kann. Zu diesen Entscheidungen werden neben den Regeln und den angeforderten Rechten auch die Subjekt- und Objekt-Attribute hinzugezogen. Eine Berechtigungsregel kann entweder so gestaltet sein, dass sie vor der Ausübung eines Rechts abgefragt wird, und diese gegebenenfalls verhindert[108], oder dass sie kontinuierlich während der Ausübung eines Nutzungsrechts geprüft wird. Hierdurch könnte etwa in Echtzeit auf die Zurücknahme von Credentials reagiert werden oder die Nutzung eines multimedialen Inhalts zu einem definierten Zeitpunkt unterbrochen werden.

Verpflichtungen definieren Auflagen, die der Nutzer eines Inhalts entweder vor der Ausübung seines Rechts oder währenddessen erfüllen muss. Das Beispiel einer vorgelagerten Verpflichtung wäre das Lesen und Bestätigen von Geschäftsbedingungen oder das Hinterlegen eigener Daten. Verpflichtungen die während der Nutzung anfallen würden, mögen etwa das Betrachten von eingestreuten Werbenachrichten während einer Mediennutzung beinhalten. Hilty et al. unterscheiden Verpflichtungen zum einen nach ihrer zeitlichen Wirksamkeit (begrenzt oder dauerhaft), zum anderen nach der Beobachtbarkeit durch den Betroffenen (beobachtbar oder nicht-beobachtbar)[109].

Schließlich sind es die Bedingungen, die den letzten Faktor für die Nutzungs-entscheidungen darstellen. Anders als Verpflichtungen sind sie nicht an die Objekte gebunden. Bedingungen prüfen die Umgebung des Systems, und können durch deren Rückmeldung die Entscheidungen zur Rechtegewährung beeinflussen. Subjekt- und Objekt-Attribute können zwar genutzt werden, um zu ermitteln, welche Bedingung zu prüfen ist, aber sie beeinflussen nicht deren Rückmeldung. Beispielsweise kann eine Bedingung lauten, dass der Zugriff zu bestimmten Daten nur zu normalen Geschäftszeiten erteilt werden soll. Diese Regel soll jedoch nur für Sachbearbeiter, nicht aber für Administratoren gelten. In diesem Sinne ist die Tageszeit die zu prüfende Bedingung, und die Rolle als Sachbearbeiter oder Administrator ein Attribut des Subjekts.

[108] Dies entspricht der Prüfung bei Klasssichen Access Control Modellen.
[109] vgl. [HiBP05]

Gerade die Möglichkeit, Attribute durch Nutzung (und nicht nur durch Eingriffe der Administratoren) verändern zu können, sowie Berechtigungen und Verpflichtungen zur Laufzeit der Ausübung eines Rechts und nicht nur davor auswerten zu können, verleiht $UCON_{ABC}$ die Fähigkeit, neben den Access Control-Varianten auch DRM-Konzepte zu modellieren[110]. Neuere Implementierung auf der Basis von Usage Control beschäftigen sich so auch vornehmlich mit der Einschränkung der künftigen Nutzung von weitergegebenen Inhalten[111].

Abschließend zeigt Abbildung 9 den Steckbrief der klassischen Access Control. Die Policies werden durch den Betreiber der Datenbank, also zumeist den Verarbeiter, formuliert und kontrolliert. Erst bei der Ablage der Daten im Datenspeicher erfolgt die Verknüpfung mit den Policies, die grundsätzlich den situationsbedingten Zugriff auf die Daten präventiv regeln, durch das Fehlen der Modellierungsmöglichkeit für Verpflichtungen aber keine echte Verwaltung des Lebenszyklus bieten. Aktualisierungs- und Löschverpflichtungen müssen außerhalb des Verfahrens implementiert werden. Durch die Protokollierung der Zugriffe ermöglichen Access Control-Verfahren die detektivische Auswertung. Den Schwachpunkt bei der Verarbeitungskontrolle wie auch der Dokumentation stellt die Administratoren-Lücke dar: Da die Policies und die Mechanismen zur Durchsetzung im Kontrollbereich des Verarbeiters liegen, gibt es bei diesem immer auch Mitarbeiter mit den Möglichkeiten, das Verfahren mittels Administratorenrechten zu kompromittieren.

Klassische Access Control

Policy-Management

Formulierungsort der Policies	Beim Verarbeiter
Perspektive der Policy-Formulierung	Interne Policies des Verarbeiters
Management von Änderungen	Migration auf die neue Policy
Sprachen zur Formulierung	SAML, WSPL, XACML
Bindung an die Daten	Technisch nicht umgehbar (aber Admin-Lücke)

Durchsetzung der Policies

Zeitpunkt / Ort	Beim Betroffenen	Beim Verarbeiter	Bei Dritten
Präventiv, vor Übermittlung			
Präventiv, vor Verarbeitung		●	
Detektivisch, nach Verarbeitung		●	
Nur Willenserklärung			

Wirkung im Daten-lebenszyklus

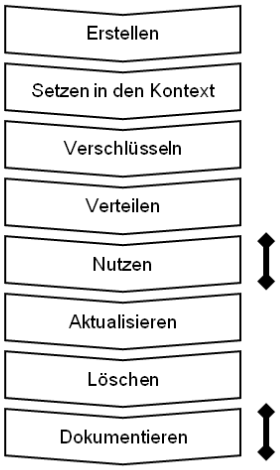


Abbildung 9: Steckbrief Klassische Access Control

[110] vgl. [ZPPS04], [ZPPS05]

[111] Demsky zeigt eine Linux-basierte Implementierung, die das Vorhandensein einer vertrauenswürdigen Systemumgebung und eine unveränderte Laufzeitumgebung des Policy-Moduls als Bedingungen für die Nutzung von multimedialen Inhalten prüft, vgl. [Dems11]. Kumari et al. fokussieren auf die Möglichkeiten eines Betroffenen, die Nutzung der durch ihn in einem sozialen Netzwerk hinterlegten Daten durch entsprechende Implementierung von Usage Control im Web Browser zu beschränken, vgl. [KPPK11].

3.3 Trusted Computing

Usage Control zeigt bereits, dass sich Access Control und Digitales Rechtemanagement (DRM) in eine gemeinsame Zielsetzung einbetten lassen. Es ist zu prüfen, welche Elemente des DRM und seiner bislang wichtigsten Basistechnologie Trusted Computing im Sinne des Datenschutzes eingesetzt werden können.

Access Control hat die Eigenschaft, dass sie auf den Schutz personenbezogener Daten *vor* deren Offenlegung gegenüber einem Verarbeiter fokussiert und die Offenlegung selbst möglichst restriktiv zu gestalten trachtet. Für Daten, die einmal offengelegt wurden, kann es keinen absoluten Schutz vor Missbrauch geben. Nur Systeme, die als Aufgabe rein statistische Auswertungen haben, könnten nach der Durchführung ihrer Auswertungen die personenbezogenen Daten löschen, ohne dass eine Person sie sehen kann. In allen anderen Fällen wird ein verarbeitendes System auch eine Schnittstelle zum Zugriff von Menschen auf die Datenbasis bereithalten und diese können immer auch missbräuchliche Absichten hegen. Im einfachsten Fall können sie Daten, mit denen sie im Rahmen ihrer Tätigkeit in Verbindung kommen, drucken, vom Bildschirm abfotografieren, oder sich schlicht merken. Dieser Schritt aus der digitalen in die analoge Welt ist praktisch nicht zu verhindern und kann immer in Missbrauch münden[112].

Allerdings erschwert das analoge Vorliegen von Daten auch den Missbrauch. Besonders wenn es um die massenhafte Verarbeitung von Daten geht, ist die digitale Form günstiger zu handhaben. Findet die Verarbeitung also digital statt, kann sie auch durch technische Mittel gesteuert werden. Die hauptsächlich diskutierten Ansätze hierzu heißen „Trusted Computing“ [113] und darauf aufsetzende Implementierungen des „Digital Rights Management“ (DRM) [114] und „Enterprise Rights Management“ (ERM)[115].

3.3.1 Grundlagen des Trusted Computing

Trusted Computing hat die Aufgabe, „sichere“ IT-Komponenten zur Verfügung zu stellen. Kuhlmann betont, dass es dabei hilfreich sei, Software nicht primär als Programmcode zu betrachten, sondern als maschineninterpretierbare Verkörperung von menschlichen Absichten und Erwartungshaltungen[116]. Software vergegenständliche also die Intention ihres Entwicklers und sei Gegenstand von Benutzererwartungen. Letztere beinhalten neben den offenbarten und dokumentierten Absichten des Entwicklers auch die implizite Annahme, dass Software in aller Regel das und nur das tue, was den Benutzererwartungen und dem vom Hersteller offenbarten Zweck entspräche.

[112] vgl. [BeGü04] zu den Optionen, Ausgabesignale auf verschiedenen Wegen mitzuschneiden.
[113] [PoRe08]
[114] u.a. [FrKa04], [Främ05]
[115] u.a. [Bitz05], [Bran05], [KhHe07]
[116] vgl. [Kuhl04a]

Der Autor definiert Sicherheitsbrüche als Verstöße gegen die oben geschilderten Erwartungen, etwa wenn IT-Komponenten durch unvorhergesehene Eingaben oder Veränderungen ihrer Laufzeitumgebung zu Aktionen veranlasst würden, die außerhalb ihres normalen Aktionsradius lägen.

Sichere IT-Komponenten sollen folglich mindestens den drei folgenden Erwartungen entsprechen:

- Ihr Verhalten entspricht der Intention des Entwicklers ebenso wie den begründeten Erwartungen des Benutzers.
- Sie entsprechen der impliziten Erwartung des Benutzers, dass sie auf dieses Verhalten beschränkt sind[117].
- Sie können nicht dazu gebracht werden, diese Beschränkungen (das heißt, das Verhalten nach der Intention des Entwicklers) zu umgehen.

Diese Punkte machen deutlich, in welchem Sinne Trusted Computing bei der Sicherung von übermittelten personenbezogenen Daten gegen Missbrauch helfen kann: Das Verhalten der empfangenden und verarbeitenden IT-Komponenten wäre dem Dateneigner bekannt, bevor er sich entschlösse, seine Daten zu übermitteln. Er könnte sich also darauf verlassen, dass die Empfängersysteme nur die vereinbarten Aktionen zur Nutzung seiner Daten durchführen, dass sie keine zweckfremde Verarbeitung vornehmen und dass sie auch nicht dazu gebracht werden können, den Missbrauch der Daten zuzulassen. Eine wesentliche Problemstellung wird dabei auch deutlich: Der Dateneigner müsste selbst über umfassenden Sachverstand verfügen, um die technischen Spezifikationen der beteiligten IT-Komponenten initial gegenüber seinen eigenen Erwartungen zu prüfen. Alternativ müsste er sich auf das Urteil einer vertrauenswürdigen dritten Partei oder eines Netzwerks (Web of Trust) verlassen, die an seiner statt die Prüfung vornehmen.

Koenig und Neumann weisen darauf hin, dass es beim Trusted Computing neben systembezogenen Aspekten wie dem sicheren Hochfahren von Systemen auch um die Begrenzung des Zugangs zu Daten geht, die in vertrauenswürdigen Systemumgebungen erzeugt, bearbeitet oder gespeichert werden. So könne beispielsweise sichergestellt werden, dass Dokumente, die auf Rechnern einer Behörde oder eines Unternehmens erstellt wurden, auch nur auf Rechnern derselben Behörde oder desselben Unternehmens geöffnet werden können. Auf diese Weise ließen sich zu einem gewissen Grad sensible Informationen vor nicht-autorisierter Weitergabe schützen – etwa an fremde Mächte, die Konkurrenz oder die Presse[118]. Führt man diesen Ansatz weiter, sind Szenarien realisierbar, in denen Datensätze innerhalb eines Unternehmens auch nur von bestimmten Applikationen und in diesen nur von festgelegten Funktionsbausteinen verarbeitet werden können. Für die Integrität dieser Funktionsbausteine sorgten die Forderungen des Trusted

[117] Diese Erwartung bezieht sich insbesondere auf den Ausschluss „trojanischer Pferde“, die versteckten Schadcode mit nützlichen Inhalten verbinden. Ebenso kann als implizite Benutzererwartung aber auch aufgefasst werden, dass Sicherheits-Komponenten frei von versteckten Hintertüren (Backdoor-Zugängen) sind.

[118] vgl. [KoNe04a]

Computing. Die Funktionsaufrufe wiederum wären an die Verarbeitungszwecke gebunden, für die sie programmiert wurden. Diese Restriktion ist zweifelsohne datenschutzfreundlich, aber nicht perfekt. Brandl und Rosteck verweisen mit Recht darauf, dass Trusted Computing zwar Authentifizierung und Beglaubigung der Plattform, nicht aber des Anwenders bietet[119]. In böswilliger Absicht sind Daten auch dann noch zu missbrauchen, wenn ihre technische Nutzung weitgehend eingeschränkt wurde. Daher kann Trusted Computing detektivischen Datenschutz keinesfalls ersetzen. Es ergänzt ihn vielmehr.

3.3.2 Trusted Platform

Zur Gestaltung und Standardisierung des Trusted Computing ist das maßgebliche Gremium die „Trusted Computing Group“[120] (TCG), der zahlreiche führende Hard- und Software-Hersteller angehören, unter ihnen Intel, AMD, Infineon, HP, Lenovo, Microsoft und Oracle. Die TCG hat gemäß der formulierten Erwartungen den Standard für eine hardwarebasierte Unterstützung für das Trusted Computing definiert, der im Jahre 2009 als ISO-Standard übernommen wurde (ISO/IEC 11889, Teile 1 bis 4)[121]. Das Kernelement ist das „Trusted Platform Module“ (TPM), ein Bauelement für Rechner, das gegen physische Manipulation gesichert ist und unter anderem folgende Fähigkeiten aufweist[122]:

- Überwachung der Systemintegrität, also die Prüfung, dass festgelegte Systemkomponenten einem einmal festgelegten Status entsprechen,
- effizientes Durchführen kryptographischer Berechnungen (zum Beispiel RSA-Verschlüsselung, Hashing),
- Erzeugung und nicht-flüchtige Speicherung von kryptographischen Schlüsseln,
- hardwarebasierte Erzeugung von Zufallszahlen (Rauschgeneratoren),
- Test auf eigene Integrität (Selbsttest).

In den 90er Jahren des vergangenen Jahrtausends von Siebert et al. noch als Vision einer „Secure Processing Unit“ skizziert[123], werden TPM nun schon seit einigen Jahren in der Mehrzahl neuer PCs und Laptops verbaut und erfahren somit bereits einen hohen Verbreitungsgrad. Im Jahr 2008 wurden laut einer Studie des Marktforschungsinstituts IDC etwa 80 Millionen Laptops und Desktop-Systeme mit TPM aufgeliefert. Nach Schätzungen kann bis zum Jahr 2018 mehr als 80 Prozent der PC-Client Basis des Internet mit TPM ausgestattet sein[124].

[119] vgl. [BrRo04], [BrCC04]
[120] http://www.trustedcomputinggroup.org (Zugriff: 18.12.2011).
[121] http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html (Zugriff: 18.12.2011).
[122] vgl. [PoRe08]
[123] [SiBW95]
[124] vgl. [PoRe08]

3.3.3 Rechtemanagement auf Basis des Trusted Computing

Digital Rights Management (DRM) stammt aus der Forschung um die Sicherung von elektronischen Medien gegen ungerechtfertigtes Vervielfältigen und Nutzen (Multimedia-Piraterie)[125] und steht in diesem Zusammenhang in der Kritik, eben keine besonders datenschutzförderliche Technik zu sein[126]. Eine Studie des Unabhängigen Landeszentrums für Datenschutz Schleswig-Holstein[127] stellt beispielsweise dar, wie DRM-Implementierungen den Schutz des Datenobjekts gegen unberechtigte Nutzung in den Vordergrund stellen, dabei allerdings das Nutzungsverhalten des Käufers ausforschen sowie die Ausübung seiner erworbenen Nutzungsrechte signifikant behindern[128]. Auch ein „schwaches" DRM, das nach der Definition von Schmidt et al. nicht den Kopierschutz zum Ziel hat, sondern die Verfolgung der Verbreitung von Inhalten[129], lässt diese Kritik unbenommen. Es kann durchaus ein Ziel von DRM sein, eine möglichst große Verbreitung von Inhalten nicht zu behindern, sondern zu fördern[130], sehr wohl aber diese Verbreitung zu dokumentieren und die Nutzung, das heißt das Abspielen des Inhalts nach den Nutzungsbedingungen des Medieneigners, zu kontrollieren. Neutral gesagt steht im Mittelpunkt eines DRM-Systems, wie die Beziehung zwischen Identität (von Personen, Geräten, Inhalten), Inhalt selbst und Regeln (verständlich für Mensch und Maschine) gehandhabt wird[131]. Dabei gilt, dass mit DRM-Systemen erstellte digitale Objekte, unabhängig davon auf welchen Systemen sie sich befinden, weiterhin der Kontrolle des jeweiligen Autors unterstehen und nicht dem Besitzer des Systems[132].

Enterprise Rights Management (ERM) ist dem DRM durchaus verwandt und nutzt zum Teil dessen technische Grundlagen. Dennoch genießt es aufgrund seiner ursprünglichen Ausrichtung auf den Informationsschutz innerhalb von Unternehmen und der Tatsache, dass die eher emotional diskutierten Aspekte des Urheberrechts von Multimedia-Produkten weniger im Vordergrund stehen, einen besseren Ruf als DRM[133]. Wo DRM insbesondere gegen die umfangreichen Missbrauch von Multimedia-Inhalten wirken muss, also ein Massenproblem zu bekämpfen sucht, muss ERM jeden einzelnen Vorgang exakt behandeln. Hier geht es um den Schutz von Geschäftsgeheimnissen, bei denen im Gegensatz zu DRM der absolute Schutz jeder Transaktion und ebenfalls in stärkerem Maße die Verfolgbarkeit von Missbrauch im Vordergrund steht.

---

[125] vgl.[RoTM02], sowie [Bech04] zu den Schutzmechanismen des DRM im Detail.
[126] vgl. [Büll04]
[127] vgl. [BiGr06]
[128] „The main problem Trusted Computing faces is – Trust." stellt Markus Hansen dazu fest [Hans04].
[129] vgl. [ScTW04]
[130] sog. „Superdistribution", kommerziell erstmals etabliert in Form des Cryptolope in IBMs Produkt infoMarket [IBM96].
[131] vgl. die Definition von Serrao und Marques in [SeMa04].
[132] Frei übersetzt nach [Ande03].
[133] vgl. [Bran05]

3.3.4 Einsatz für den Datenschutz

DRM-Techniken wurden unstrittig zum Schutz der kommerziellen Interessen von Inhalteanbietern erfunden. In ihrer Zielstellung, die Rechte eines Dateneigners auf dem Rechner eines Nutzers durchzusetzen[134], wurden sie aber in den Jahren nach der Jahrtausendwende zunehmend mit ihrem Potential zum Schutz personenbezogener Daten diskutiert[135]. Goldberg führt beispielhaft die Umwidmung von DRM-Techniken zum Schutz medizinischer Daten an, so dass ein Betroffener etwa seine Daten an einen Anbieter medizinischer Dienstleistungen herausgeben würde, aber dieser sie nicht ungebeten etwa an das nächste Reformhaus weitergeben könne[136].

Bizer kommt nach Untersuchung der Angreifermodelle und der Schutzmaßnahmen durch ein hardwaregestütztes DRM zu dem Schluss: „Gerade für den Datenaustausch mit Externen könnte Trusted Computing den einzig gangbaren Weg darstellen, um die Integrität einer entfernten, nicht direkt zugänglichen Plattform zu überprüfen. Damit wäre ein nicht unerheblicher Sicherheitsgewinn beim Austausch von Daten mit Externen zu erzielen.“[137]

Wünschenswert scheint ein „Privacy Digital Rights Management“ (Privacy-DRM), das sich analog zu den Zielen bestehender DRM-Systeme für multimediale Daten dadurch auszeichnen würde, dass es dem Dateneigner die Hoheit über jede Art der Nutzung seiner Daten gewährt. Das bedeutet, er übt volle Kontrolle über seine Inhalte, von der ersten Bereitstellung über alle Übertragungen bis zur letzten Nutzung durch einen Verarbeiter und die Löschung aus.

DRM-Systeme zum Schutz medialer Inhalte wurden in der Vergangenheit eingeführt und sind technisch effektiv[138], sofern es einem Medien-Anbieter, also dem Dateneigner gelingt, ausreichend Marktmacht zu versammeln. Dann kann er die notwendigen Standardisierungen in der Hard- und Softwareausstattung herbeiführen, die für die Nutzung der geschützten Inhalte grundlegend sind. Die in der Trusted Computing Group

---

[134] vgl. [BiGW06]

[135] „[...] we perceived a *Digital Rights Management* (DRM) aspect of an unusual kind. In the standard DRM scenario, a large data producer seeks to control use of its data by multiple ‘small’ users. We see the possibility to turn this around and enable individual users to control with confidence how their data is to be used by a powerful authority (‘Big Brother’).“ [IlSm02], „Ein gemeinsames Grundproblem von DRM-Systemen ist, dass sie die „Sicherheit“ von medialen Inhalten auf Systemen gewährleisten sollen, die von anderen administriert werden.“ [Hans06]

[136] „Some people have (sometimes half-jokingly) suggested that Digital Rights Management techniques from the online music arena could be flipped on their heads to help us out here; a consumer would protect his personal data using a DRM technique, so that it could be used only in the ways he permits, and could not be passed from health care provider to his health food salesman.“ [Gold02]

[137] [Bitz05]

[138] Allerdings gilt die Durchsetzung von DRM zumindest in der Musik-Distribution als weitgehend am Widerstand der Nutzer gescheitert und die entsprechenden Maßnahmen wurden von vielen großen Anbietern eingestellt. Für Filme, elektronische Bücher und Computerspiele ist die Durchdringung aber signifikant.

versammelten Unternehmen einerseits und das durch die Firma Apple andererseits proprietär eingeführte DRM[139] sind Beispiele dafür.

Dies wirft die Frage auf, ob es Betroffenen der Verarbeitung personenbezogener Daten gelingen kann, eben solche Marktmacht zu vereinen wie die Anbieter von Multimedia-Inhalten. Für individuelle Betroffene lässt sich die Frage sofort verneinen. Doch auch für die Durchsetzung durch organisierte Gruppen ist wenig Potential zu erkennen. Haben sich die Standardisierungsbemühungen der TCG aus einem gemeinsamen monetären Interesse der beteiligten Unternehmen ergeben, der Eindämmung des Raubkopierwesens, so finden sich demgegenüber heterogene Interessenlagen bei den Betroffenen von Personendatenverarbeitung. Während der eine den Komfort schätzt, den ihm vernetzte Datensammlungen bieten, kämpft der andere für starke Nutzungskontrolle. Hier wiederum ist der eine bereit, Investitionen bis zu einem bestimmten Maß zu tätigen, während der andere eine zunächst kostenlose Lösung bevorzugt. Die Rechenmodelle der Multimedia-Anbieter, welche Investitionen für die Einführung eines DRM durch die resultierende Begrenzung der Einnahmeausfälle kompensiert werden, ist für den Bereich des Datenschutzes nicht anwendbar, da dessen Wert personenindividuell bemessen wird.

Es bliebe also die gesetzliche Regulierung. Hier sind zurzeit keine Ansätze erkennbar, ein Privacy-DRM verpflichtend einzuführen. Nationale Bemühungen wären ohnehin in Anbetracht eines grenzüberschreitenden Datenaustauschs sinnlos. Und internationalen Initiativen fehlt die gemeinsame Bewertung der Datenschutzproblematik. Zu heterogen sind auch hier die Ziele der jeweiligen Regierungen im Spannungsfeld zwischen Wirtschaftsfreundlichkeit und Bürgerschutz.

Man muss also schließen, dass unter den gegebenen Bedingungen ein durchgängiges und technisch durchgesetztes Digital Rights Management für personenbezogene Daten zwar theoretisch realisierbar, aber praktisch nicht umsetzbar ist.

Im Steckbrief der Abbildung 10 fällt insbesondere der hohe Abdeckungsgrad im Datenlebenszyklus auf, der einem funktionsfähigen System ein starkes Datenschutzniveau nahelegen würde. Gleichzeitig wäre die aus Betroffenen-Sicht zu präferierende Kombination gegeben, bei der die Policy-Kontrolle beim Dateneigner läge, die zuverlässige präventive Durchsetzung jedoch beim Verarbeiter.

[139] vgl. [Venk07] zu Apples DRM-System „FairPlay“.

Trusted Computing

Policy-Management

Formulierungsort der Policies	Beim Dateneigner
Perspektive der Policy-Formulierung	Eigner-Präferenzen, Auflagen für Sekundär-Nutzer
Management von Änderungen	Fortführen der Beziehung mit den alten Policies
Sprachen zur Formulierung	ODRL, XrML
Bindung an die Daten	Technisch nicht umgehbar, Sticky Policies

Durchsetzung der Policies

Zeitpunkt / Ort	Beim Betroffenen	Beim Verarbeiter	Bei Dritten
Präventiv, vor Übermittlung			
Präventiv, vor Verarbeitung		●	
Detektivisch, nach Verarbeitung			
Nur Willenserklärung			

Wirkung im Daten-lebenszyklus

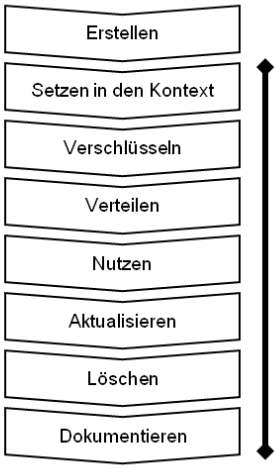


Abbildung 10: Steckbrief Trusted Computing

3.4 Hippokratische und aktive Datenbanksysteme

In Hippokratischen Datenbanken werden die Policies zusammen mit den Daten gespeichert und bei jeder Abfrage geprüft. Eine Hippokratische Datenbank verpflichtet sich, Daten nicht preiszugeben, für deren Abfrage keine hinreichende Rechtfertigung vorliegt.

3.4.1 Herkunft

Hippokratische Datenbanken wurden erstmals im Jahre 2002 vom IBM-Forschungszentrum Almaden vorgeschlagen[140]. Sie verdanken ihren Namen der Analogie, die ihre Erfinder zu dem Hippokratischen Eid der Medizin[141] sahen: „Verantwortungsvoller Umgang mit dem Schutz der verwalteten Daten ist das wesentliche Grundprinzip Hippokratischer Datenbanken.“ [142] Die Datenbank überlässt die datenschutzkonforme Behandlung der in ihr abgelegten Daten nicht den verarbeitenden Programmen oder externen Prüfungen, sondern verpflichtet sich selbst zu deren Schutz. Das erfolgt unter

[140] [AKSX02]

[141] Der Eid des Hippokrates, entstanden um 400 v.Chr., regelt als grundsätzliche Formulierung ärztlicher Ethik unter anderem auch die Schweigepflicht des behandelnden Arztes und wird daher oft als das älteste Datenschutz-Gesetz bezeichnet (vgl. etwa: de.wikipedia.org/wiki/Hippokratischer_Eid; Zugriff: 21.03.2012).

[142] Im Original [BaSr03]: „Hippocratic databases include responsibility for the privacy of data they manage as a fundamental tenet.“

Einsatz von Technik, die bei der Verwaltung von Daten die Bestimmungen zu deren Offenlegung (Datenschutz, Datensicherheit, gesetzliche Regelungen) respektiert.[143]

Die Architektur basiert auf den Erkenntnissen mehrstufig sicherer Datenbanksysteme. Diese erlauben bereits, für die zu speichernden Daten verschiedene Sicherheitsstufen zu definieren, z.B. in abnehmender Stringenz: Top Secret, Secret, Confidential und Unclassified[144]. Abfragen haben (in Abhängigkeit von ihrer Herkunft) verschiedene Sicherheitsstufen, die gegen die Sicherheitsstufe der abgefragten Daten geprüft werden. Beziehen sich Abfragen auf Daten, die einer anderen Sicherheitsstufe unterliegen, finden die Prinzipien des „No-read-up“ (Eine Abfrage kann nicht Datenfelder einer höheren Sicherheitsstufe lesen) und „No-write-down“[145] (Abfragen dürfen keine Datenbank-Einträge einer niedrigeren Sicherheitsstufe anlegen, weil dies die Restriktionen umgehen würde, denen niedriger eingestufte Abfragen korrekterweise unterliegen) Anwendung.

Die mehrstufig sicheren Datenbanksysteme weisen damit bereits Eigenschaften auf, die den datenschützerischen Kern von Hippokratischen Datenbanken darstellen:

- Metadaten beschreiben das Schutzbedürfnis der gespeicherten Inhalte gemäß einem festgelegten Vokabular.
- Die Metadaten werden gemeinsam mit den Inhalten und in direktem Bezug auf diese gespeichert.
- Abfragen erfolgen ausschließlich über definierte Schnittstellen. Diese erlauben den Zugriff auf Daten nur, wenn die entsprechenden Metadaten in der Prüfung gegen den Kontext der Abfrage dies zulassen.

Die starre Definition der Sicherheitsstufen schränkt die Flexibilität dieser Datenbanksysteme gegenüber komplexen Sachverhalten stark ein. Hippokratische Datenbanken setzen hingegen formale Sprachen ein, um individuell die Schutzbestimmungen festzulegen.

3.4.2 Grundprinzipien

Die zehn grundlegende Prinzipien Hippokratischer Datenbanken, im Jahre 2004 formuliert, lauten[146]:

1. „Purpose Specification“: Der Zweck, zu dem die personenbezogenen Daten erhoben wurden, muss mit ihnen gespeichert werden.

[143] „[…] which respects the disclosure policies (e.g. privacy, security, legislative compliance policies) of data it manages.“ [Alma04]
[144] [AKSX02]
[145] Die Prinzipien des „No-read-up“ (NRU) und „No-write-down“ (NWD) entstammen dem Zugriffs-kontrollmodell nach Bell-LaPadula, vgl. [Ecke04] S.266ff, [Trce06].
[146] [Alma04]

2. „Consent“: Der Verarbeitungszweck, der mit den personenbezogenen Daten gespeichert wird, unterliegt der Einwilligung des Betroffenen.
3. „Limited Collection“: Der Umfang der erhobenen Daten ist auf das minimal Notwendige zur Wahrnehmung des spezifizierten Zwecks zu beschränken.
4. „Limited Use“: Das Datenbanksystem darf nur Abfragen ausführen, deren Zweck sich mit dem der ursprünglichen Datenerhebung deckt.
5. „Limited Disclosure“: Gespeicherte personenbezogene Daten dürfen nur weitergegeben werden, wenn dies zu einem Zweck erfolgt, zu dem die Einwilligung des Betroffenen vorliegt[147].
6. „Limited Retention“: Personenbezogene Daten sollen nur solange gespeichert werden, wie es zur Wahrnehmung des zur Erhebung maßgeblichen Zwecks notwendig ist.
7. „Accuracy“: Die gespeicherten Daten müssen korrekt und aktuell sein.
8. „Safety“: Die gespeicherten personenbezogenen Informationen müssen durch geeignete Sicherheitsmaßnahmen gegen Diebstahl oder sonstigen Missbrauch geschützt werden.
9. „Openness“: Dem Betroffenen muss der Zugang zu allen über ihn in der Datenbank gespeicherten Informationen möglich sein.
10. „Compliance“: Ein Betroffener muss die Möglichkeit haben, die Einhaltung dieser Prinzipien zu überprüfen. Dementsprechend muss das Datenbanksystem Mechanismen bereithalten, die eine Prüfung unterstützen.

Diese Prinzipien finden ihre Entsprechung in den international anerkannten Richtlinien, etwa den von der Organisation für wirtschaftliche Zusammenarbeit und Entwicklung formulierten, in leicht veränderter Gewichtung[148].

3.4.3 Funktionsweise

IBM beschreibt Hippokratische Datenbanken als ein „Technology Set“, das die folgenden Komponenten beinhaltet[149]:

- Aktive Durchsetzung von vereinbarten Datenschutzregelungen durch Abgleich der Datenschutzpolicies und der hinterlegten Benutzerpräferenzen,
- Compliance Auditing,

[147] vgl. auch [MaMZ05], die durch starke Dekomposition der Daten sogar ein „Minimum Disclosure“ erreichen wollen.
[148] „Consent“ und „Limited Collection“ finden sich in den Richtlinien der OECD im „Collection Limitation Principle“, „Limited Use“ und „Limited Disclosure“ im „Use Limitation Principle“ vereint. Die anderen Prinzipien finden ihre direkte Entsprechung, wenn auch in veränderter Reihenfolge in [OECD02].
[149] [Alma04]

- „Sovereign Information Integration“, also Informationsaustausch zwischen zwei Parteien, wobei beiden nur die benötigte Schnittmenge der Informationen preisgegeben wird[150],
- datenschutzfreundliches Data Mining durch Verfälschen von Einzeldaten unter Beibehaltung der statistischen Relevanz[151],
- Watermarking der Datenbankinhalte um Datendiebstahl zu verhindern und die Herkunft gestohlener Daten reidentifizieren zu können,
- „Order Preserving Encryption“, bei der die Verschlüsselung so gewählt wird, dass die Chiffrate einer vorgegebenen numerischen Verteilung und Reihenfolge unterliegen, und damit auch über verschlüsselte Datenbestände Operationen durchgeführt werden können, ohne eine Entschlüsselung vorzunehmen.[152]

3.4.4 Beschreibung der Datenschutzregelungen

Zur Beschreibung von Datenschutzregelungen sehen die Schöpfer der Hippokratischen Datenbanken EPAL (Enterprise Privacy Authorization Language), eine formale XML-basierte Sprache zur Definition von Datenschutzregelungen, die IBM im Jahre 2003 als Vorschlag[153] dem W3C vorgelegt hat. EPAL erweitert die Aushandlungssprache P3P um Möglichkeiten zur Durchsetzung der Vereinbarungen und zielt dabei insbesondere auf die Anwendung in oder zwischen Firmennetzen[154]. In EPAL formulierte Datenschutzpolicies können in hippokratischen Datenbanken automatisch durchgesetzt werden. Im Gegensatz zu P3P besitzt EPAL kein Standard-Vokabular für Datenschutzregelungen. Dieses wird in einem ersten Schritt definiert. Zu einem solchen Vokabular gehören nach der Definition von May[155]:

- Zulässige Verarbeitungszwecke,
- Benutzerkategorien (Rollen),
- Datenkategorien und Datenfelder,
- Erlaubte Aktionen,
- Verpflichtungen („Obligations“).

Dann kann die eigentliche Datenschutz-Policy definiert werden. In ihr können Bedingungen im XACML-Format beschrieben werden, die zusammen mit den

[150] Ausführlicher widmen sich den Möglichkeiten des „Sovereign Information Sharing“ Agrawal et al. in [AgAS04].
[151] Zum „Privacy preserving data mining“ vgl. auch [BaSr03] S.116f.
[152] So sind Operationen wie Sortieren, Zählen, Finden des Minimums und des Maximums ohne Entschlüsselung möglich, vgl. [AKSX04].
[153] vgl. www.w3.org/2003/p3p-ws/pp/ibm3.html und www.w3.org/Submission/2003/07/Comment (Abfrage: 21.03.2012).
[154] vgl. [May04] für Erläuterung und ein ausführliches Beispiel zu EPAL 1.2.
[155] [May04]

festgelegten Vokabeln und den in der EPAL-Spezifikation hinterlegten Operatoren (z.B. Vergleichsoperatoren zur Prüfung von Feldwerten und die Entscheidungsoperatoren „accept“ und „deny“) zum Aufbau des Regelwerks verwendet werden können.

3.4.5 Bindung an die Daten

Die beschriebenen Policies werden durch einen EPAL-Parser[156] als Attribute in Tabellen der Datenbank geschrieben. Zuvor findet jedoch zusätzlich ein Abgleich der Präferenzen des Betroffenen mit den (organisationsintern definierten) Policies statt, beispielsweise um Einverständniserklärungen oder bestimmte Vorstellungen zu den Aufbewahrungszeiten zu berücksichtigen. Die Präferenzen können beispielsweise in den Sprachen APPEL oder XPref formal ausgedrückt werden.

Zu jedem Datenfeld, das in der Hippokratischen Datenbank für personenbezogene Daten vorgesehen ist, werden in zwei zusätzlichen Tabellen („Privacy-Policies Table“ und „Privacy-Authorizations Table“) Einträge angelegt, die dem Datenfeld zuordnen, zu welchen Zwecken welche externen Empfänger oder internen Bearbeiter Zugriff erhalten dürfen und welcher Aufbewahrungszeitraum für die dort gespeicherten Daten jeweils gilt. Dabei können je nach Ausgestaltung der Systemlandschaft die potentiellen externen Empfänger und die internen Bearbeiter durch Rollenbezeichnungen (z.B. „Vertriebsmitarbeiter“, „Spediteur“) oder technische Daten (z.B. der Name des aufrufenden Dienstes, IP-Adressräume) repräsentiert werden.

Diese Meta-Tabellen legen Restriktionen für die Datenfelder (Spalten) der eigentlichen Datentabellen fest. Zusätzlich sehen Hippokratische Datenbanken auch die Aufnahme eines neuen Feldes für den Verarbeitungszweck in jede relevante Tabelle vor. Auf diese Weise können die speziellen Bedürfnisse jedes einzelnen Betroffenen auf der Ebene seiner individuellen Datensätze (Zeilen) berücksichtigt werden. Erst die Kombination der Zweck-Festschreibung auf der Ebene der Felddefinitionen mit der direkten Zweck-Festschreibung für jeden einzelnen Datensatz macht das für Hippokratische Datenbanken charakteristische Enforcement auf Zellenebene möglich.

3.4.6 Durchsetzung der Regelungen

Die Enforcement-Komponenten des Datenbanksystems setzen an verschiedenen Stellen der Verarbeitung an:

[156] Einen Überblick über die technische Architektur bietet http://www.almaden.ibm.com/cs/projects/iis/hdb/Hippocratic_Arch_Overview_7_02.ppt (Abfrage: 21.03.2012). In frühen Entwürfen von 2002 wird noch der generische Begriff „Privacy Metadata Creator“ verwendet. EPAL wurde erst später zur Beschreibung der Datenschutz-Policies vorgeschlagen.

Vor der Ausführung einer Abfrage wird geprüft, ob der interne Bearbeiter, der die Abfrage ausgelöst hat, zu den autorisierten Benutzern für den spezifischen Zweck gehört. Es muss ein entsprechender Eintrag (Zweck, Benutzer) in den Metadaten vorhanden sein. Die nächste Prüfung untersucht, ob die in der Abfrage angeforderten Datenfelder für den spezifizierten Zweck zugelassen sind. Es muss also ein passender Metadatensatz (Zweck, Tabelle, Feld) existieren.

Während die Abfrage ausgeführt wird, kommt die Prüfung auf der Ebene des einzelnen Datensatzes ins Spiel. Nur Datensätze, deren Wert im Feld „Verarbeitungszweck" mit dem Zweck der Abfrage übereinstimmt, werden zur weiteren Verarbeitung zugelassen.

Technisch wird die Durchsetzung dieser Regeln durch ein „Umschreiben" der eigentlichen Abfragen (die Aufnahme zusätzlicher Bedingungen) durch das Datenbanksystem realisiert. Alternativ könnten aufgrund der Kontextinformationen (Zweck, Bearbeiter) eingeschränkte Tabellenviews zur Verfügung gestellt werden, die den Original-Abfragen anstelle der vollständigen Tabellen zur Verfügung stehen[157]. Letztere Alternative dürfte jedoch bei umfangreichen Datenbeständen und bei vielen verschiedenen Kombinationen aus Zweck und Bearbeiter zu einem stark erhöhten Speicherbedarf für das Zwischenspeichern der zahlreichen Views und damit einhergehenden Performance-problemen führen.

Hippokratische Datenbanken sehen es als einen weiteren Teil des Enforcement an, ungewöhnliche Aktivitäten zu analysieren. Sie prüfen, ob aktuelle Abfragen mit den üblichen Abfragemustern des betreffenden Benutzers zum jeweiligen Verarbeitungszweck übereinstimmen[158].

Insbesondere zur Einhaltung von Löschfristen zeigt sich, dass Hippokratische Datenbanken immer auch aktive Datenbanken[159] sind. Die Datenbank kann sich nicht auf das Warten auf Abfragen von außen beschränken, sie muss vielmehr aus eigener Initiative regelmäßig das Eintreten bestimmter Voraussetzungen prüfen und entsprechend reagieren. Im hier geschilderten Zusammenhang ist das die Prüfung des aktuellen Systemdatums gegen den festgelegten Löschzeitpunkt eines Datensatzes oder -feldes und gegebenenfalls das Veranlassen der Löschung.

Eine Durchsetzung von Policies bereits innerhalb der Datenbank durch aktive Komponenten hat einige Vorteile. Für alle Geschäftsapplikationen, die auf die Daten zugreifen, gelten dieselben Regeln. So können Verzögerungen und Inkonsistenzen vermieden werden, die auftreten würden, wenn das Regelwerk kontinuierlich zwischen

[157] [LAER+04]
[158] vgl. [AKSX02]
[159] Aktive Datenbank-Managementsysteme (ADBMS) defniert [DiGG96]. ECA-Regeln (Event-Condition-Action) ermöglichen ereignisgesteuerte Verarbeitung, ohne dass regelmäßiges Abfragen bestimmter Zustände nötig ist. Den Fähigkeiten klassischer Datenbanken fügen sie Komponenten für die Detektion von Ereignissen, die Auswertung von Bedingungen, das Ausführen von Aktionen sowie die Möglichkeit zur Kopplung der Vorgenannten hinzu, vgl. [DiGa00], [Zimm01].

verschiedenen Applikationen synchronisiert werden müsste. Die Administration der Regeln ist sicherer und einfacher[160].

Lefevre et al. zeigen weiterhin, dass die vielfach praktizierte Anwendung von Datenschutzregeln auf den Applikationsebenen über der Datenbank weniger performant ist als die Durchsetzung direkt auf der Datenbankebene. Im ersten Fall müssten zunächst alle angeforderten Daten aus der Datenbank gelesen, an die Applikation übertragen und schließlich dort, vor der weiteren Verarbeitung, hinsichtlich der Datenschutzregeln gefiltert werden. Eine Hippokratische Datenbank hingegen erledigt die Verarbeitung der Abfrage und das Filtern nach den Schutzregeln in einem Schritt und überträgt nur die tatsächlich autorisierten Daten[161].

Die Datenbank als zentraler Speicherort sensibler Daten unterliegt in jedem Fall besonderen Anforderungen an ihre Vertrauenswürdigkeit – und die ihrer Administratoren. Es scheint also vorteilhaft, möglichst viel der Verantwortung für den Datenschutz direkt in ihr zu belassen und nicht an verschiedene, teilweise wechselnde und Änderungen unterworfene, möglicherweise sogar unbekannte Applikationen abzugeben.

3.4.7 Audits

Ein möglicher Audit von Hippokratischen Datenbanken wird nach Maßgabe des „Compliance"-Prinzips berücksichtigt. So soll die Einhaltung der Selbstverpflichtung überprüfbar gemacht werden. Der Bedarf nach einem solchen Audit kann durch Beschwerden von außen oder aufgrund interner Richtlinien einer Organisation bestehen. Während des Betriebs der Datenbank werden alle Abfrage-Kommandos gemeinsam mit dem Zeitpunkt, der Anwenderkennung des Abfragenden und dem Zweck der Operation gespeichert[162]. Durch Speicherung aller Änderungen in der Datenbasis kann zum Audit zusätzlich der Zustand der Datenbank zu jedem beliebigen Zeitpunkt der Vergangenheit rekonstruiert werden[163]. Freilich kann die Menge der zu verwaltenden Daten aus der Vergangenheit erheblichen Umfang annehmen, so dass die entstanden Backlog-Dateien mit den herkömmlichen Archivierungswerkzeugen eines Datenbanksystems nach und nach auszulagern sind. Zur Durchführung eines Audits kann die Auswertung der gespeicherten Abfrage-Kommandos, gegebenenfalls nach zusätzlicher vorheriger Rekonstruktion des zu betrachtenden Zeitpunktes, mit datenbankeigenen Mitteln, also SQL-Abfragen, erfolgen[164].

---

[160] vgl. [PoAl05]
[161] vgl. [LAER+04]
[162] vgl. [AABG+05]
[163] „During normal operation, the text of every query processed by the database system is logged along with annotations such as the time when the query was executed, the user submitting the query, and the query's purpose. The system uses database triggers to capture and record all updates to base tables in backlog tables for recovering the state of the database at any past point in time." [ABFK+04]
[164] Eine ausführliche Darstellung der Prüfalgorithmen sowie eine Diskussion über Performanceaspekte des permamenten Loggings in [ABFK+04].

3.4.8 Diskussion

Hippokratische Datenbanken könnten ihren Charme daraus beziehen, dass sie anstelle herkömmlicher Datenbanken eingeführt werden können, ohne dass Veränderungen an den nutzenden Applikationen vorgenommen werden müssten. Lefevre et al. schlagen gar eine Auslagerung der Metadaten und der „Verarbeitungszweck"-Felder vor, so dass auch bestehende Tabellen aus vorhergehenden Datenbanken ohne Änderungen übernommen werden könnten[165]. Welchen Einfluss die dann notwendigen externen Verknüpfungen in den Datenbank-Abfragen auf die Gesamtleistung des Systems haben, ist noch nicht abschließend untersucht.

Kritisch zu sehen ist die Fragestellung, ob im Falle des Eintretens einer Löschverpflichtung für bestimmte personenbezogene Daten dieser überhaupt nachgekommen werden kann. Diese Problematik ist allen Datenbanken zu Eigen, die zum Zwecke der Archivierung, Sicherung und Wiederherstellbarkeit zu einem bestimmten Zeitpunkt Daten redundant speichern. Sie tritt bei Hippokratischen Datenbanken aber aufgrund der erzwungenen umfangreichen Protokollierung von allen inhaltlichen Änderungen in besonderem Maße auf[166]. Die Frage, ob eine Löschverpflichtung von Daten zugleich die Vernichtung jeder Nachweismöglichkeit, dass diese Daten existiert haben, bedeutet, wäre zu klären.

Der Steckbrief in Abbildung 11 weist eine gegenüber der klassischen Access Control verbesserte Abdeckung des Datenlebenszyklus aus, sowie durch den Aushandlungsmechanismus einen erhöhten Einfluss des Betroffenen auf die Policy-Gestaltung. Die Problematik der Administratoren-Lücke besteht jedoch auch hier.

3.4.9 Implementierungen

Das IBM-Forschungszentrum berichtet von der Implementierung eines Prototyps für die Gesundheitsbranche, bei dem die Durchsetzung der Regelungen des „Health Insurance Portability and Accountability Act" (HIPAA[167]) im Vordergrund standen[168]. Das System ist nach Angaben des Zentrums die erste hochperformante Implementierung von zellenbezogener Zugriffskontrolle[169].

[165] [LAER+04]
[166] [AKSX02]
[167] HIPAA wurde 1996 in den USA zum Schutz von Patientendaten erlassen und fordert unter anderem policy-basierte Zugriffskontrolle für einzelne Dokumente der (elektronischen) Patientenakte; vgl. www.hhs.gov/ocr/AdminSimpRegText.pdf (Abfrage: 08.01.2012), insbesondere Part 164 – Security and Privacy, siehe ebenso [MuWa10].
[168] www.almaden.ibm.com/software/projects/iis/hdb/science.shtml (Abfrage: 08.01.2012).
[169] Weitere Implementierungsbeispiele sind beschrieben in [AABG+05], [AABG+05a].

Hippokratische Datenbanken

Policy-Management

Formulierungsort der Policies	Verhandlung zwischen Eigner und Verarbeiter
Perspektive der Policy-Formulierung	Eigner-Präferenzen, int. Policies des Verarbeiters
Management von Änderungen	Fortführen der Beziehung mit den alten Policies
Sprachen zur Formulierung	APPEL, Xpref (Eigner), EPAL (Verarbeiter)
Bindung an die Daten	Sticky-Policies, aber Admin-Lücke

Durchsetzung der Policies

Zeitpunkt / Ort	Beim Betroffenen	Beim Verarbeiter	Bei Dritten
Präventiv, vor Übermittlung			
Präventiv, vor Verarbeitung		●	
Detektivisch, nach Verarbeitung		●	
Nur Willenserklärung			

Wirkung im Datenlebenszyklus

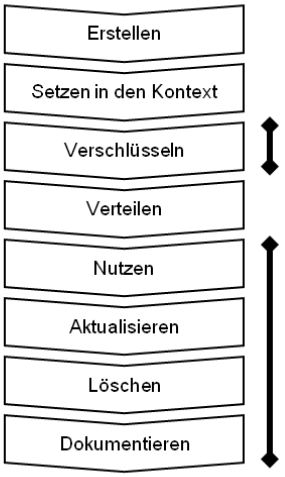


Abbildung 11: Steckbrief Hippokratische Datenbanken

3.5 Privacy Management Systeme

3.5.1 Herkunft

Parallel zu IBMs Entwicklungen an Hippokratischen Datenbanken begann die Forschung im Bereich von Privacy Management Systemen. In der von der EU beauftragten Untersuchung „Privacy Incorporated Software Agents“ (PISA) wurde der Ansatz formuliert, dass intelligente Software-Agenten den Benutzer von der Bewältigung der Informationsarbeiten entlasten sollten, indem sie mit den entsprechenden Daten und Vorgaben ausgestattet, Teilaktivitäten automatisiert durchführten[170]. Casassa Mont et al.[171] führten dieses Konzept im Rahmen des PRIME-Projekts[172] fort, das sich der Erforschung von Techniken widmete, mit deren Hilfe Betroffene und Verarbeiter den Umfang und die Bedingungen zur Nutzung personenbezogener Daten legal verbindlich aushandeln und durchsetzen können.

3.5.2 Funktionsweise

Wie auch die Hippokratischen Datenbanken beschäftigt sich der Ansatz vorwiegend mit dem Schutz von Daten innerhalb von Unternehmensdatenbanken. Auch hier werden die

[170] [BlBO03]
[171] vgl. [CaPB04] zum nachstehend beschriebenen Modell der PMS.
[172] vgl. [Reim04], Privacy and Identity Management for Europe [PRIME08].

Daten durch Verschlüsselung vor unbefugtem Zugriff gesichert. Der wesentliche Unterschied ist jedoch, dass in Privacy Management Systemen die Datenschutz-Policies nicht durch die Logik der Datenbank durchgesetzt werden. Vielmehr führen sie eine Software-Schicht zwischen dem Datenbanksystem und der Anwendung ein, das „Privacy Virtualization System“ (PVS). Dabei können herkömmliche Datenbanksysteme zum Einsatz kommen, und auch der Zugriff von Applikationen auf den Datenspeicher kann in gewohnter Art erfolgen. Nur wenn Zugriff auf die verschlüsselt abgelegten personenbezogenen Daten gefordert wird, muss das PVS einbezogen werden, um die Schlüssel zu beschaffen, die nötig sind, um an den Klartext der Daten zu gelangen. Die Eigner und Administratoren der Datenbanken verfügen also nicht über die entsprechenden Schlüssel, was eine Verbesserung der Sicherheit gegenüber den Hippokratischen Datenbanken darstellt. Auch ist es möglich, verschlüsselte Datensätze ohne Kenntnis des Schlüssels abzufragen, was etwa das Delegieren von Datenabrufen und das Weitergeben der erhaltenen Datensätze ermöglicht, ohne dass der Vertreter Einblick in die Daten nehmen könnte.

3.5.3 Policy-Beschreibung und Datenablage

Für die Ablage der geschützten Daten in der Datenbank sorgt das PVS. Es erzeugt einen symmetrischen kryptographischen Schlüssel für jeden relevanten Datensatz, verschlüsselt die Daten selbst und die zu ihrer Nutzung formulierte Policy, versieht das Chiffrat mit einer Klartextkopie der Policy und übergibt den so erstellten Datensatz an die Datenbank.

Den verwendeten symmetrischen Schlüssel verschlüsselt das PVS mit dem öffentlichen asymmetrischen Schlüssel des Privacy Management Service (PMS) und sendet ihn zusammen mit der Policy an den PMS. Danach vernichtet das PVS die lokale Kopie des symmetrischen Schlüssels und des Datensatzes im Klartext. Von da an ist nur der PMS in der Lage, die Schlüssel für die Entschlüsselung der personenbezogenen Daten in der Unternehmensdatenbank wiederherzustellen.

Die Formulierung der Policies muss maschinenlesbar erfolgen, so dass der PMS später eindeutige Entscheidungen im Sinne des Betroffenen treffen kann. Casassa Mont überlässt die Wahl der Sprache der Implementierung, schlägt aber XML beziehungsweise als dedizierte Rechtebeschreibungssprachen P3P und EPAL vor[173].

Die Ablage von Daten in der Datenbank erfolgt für Anwender in der gewohnten Art, bei SQL-basierten relationalen Datenbanken also durch INSERT- oder UPDATE-Statements. Das PVS erweitert die Befehlspalette jedoch um die Statements, mit denen der Anwender die Datenablage mit Policies verknüpfen kann, was dann im PVS die oben geschilderte Verschlüsselung auslöst.

[173] vgl. [Casa04]

3.5.4 Durchsetzung der Policies

Applikationen setzen Abfragen an die Datenbank ab, wobei diejenigen nach geschützten personenbezogenen Daten vom PVS übernommen werden. Das PVS übernimmt nun die Rolle des Vermittlers zwischen dem Anfragenden und dem PMS. Dabei liest das PVS die Policy des angefragten Datensatzes, sammelt die zum Abgleich mit der Policy benötigten Informationen und stellt sie dem PMS zur Verfügung. Dieser entscheidet über die Rückmeldung zur Abfrage, indem er die gelieferten Informationen hinsichtlich ihrer Übereinstimmung mit der Policy überprüft. Dazu gehören beispielsweise die Identität und die Credentials des Anfragenden ebenso wie erklärte Verarbeitungszwecke, aber auch Systeminformation und Daten zum bisherigen Kommunikationsverlauf mit dem Verarbeiter. Der Abgleich mit den Policies kann dann zum einen in der vollständigen Freigabe eines Datensatzes, d.h. der dafür benötigten symmetrischen Schlüssel an das PVS resultieren. Er kann zum zweiten auch zur Verweigerung der Datenfreigabe führen, wenn die Credentials nicht die Kriterien der Policy erfüllen. Oder aber es erfolgt die Freigabe eines Teils des Datensatzes, das bedeutet auch nur eines Teils der Menge symmetrischer Schlüssel, die die Felder des Datensatzes schützen. Dies ist etwa dann der Fall, wenn die Credentials des Anfragenden zwar dazu ausreichen, Name und Adresse einer Person abzufragen, aber nicht deren ebenfalls gespeicherte Religionszugehörigkeit. Das PVS bereitet die aus der Datenbank abgefragten Datensätze in jedem Fall dem Anfragenden gemäß der vom PMS freigegebenen Schlüssel in einem geeigneten View auf.

Das PMS verfügt zur Realisierung der Policy-Abgleiche über verschiedene Module zur Authentisierung der Anfragenden, zur Überprüfung von Credentials, aber auch zur Abfrage zusätzlicher Kontextinformationen. So ist es etwa möglich, Policies durchzusetzen, die eine Datenweitergabe von der Systemkonfiguration des Abfragenden abhängig machen (z.B. von der Frage ob ein Trusted Platform Module aktiv ist, oder ein Virenscanner)[174]. Die Abfrage von Daten kann auch den Kontext verändern, so dass Policy-Teile zusätzlich wirksam werden oder wegfallen, was wiederum Auswirkungen auf die Bewertung künftiger Anfragen des Verarbeiters oder auch beliebiger Anfragender haben kann.

Ein weiteres Modul des PMS ist das Audit-Modul. Es zeichnet alle Interaktionen mit Anfragenden auf, insbesondere die Herausgabe der symmetrischen Schlüssel. Durch eine Implementierung, die es vor Verfälschung sichert, kann es im Missbrauchsfall Beweismittel für forensische Analysen liefern[175].

[174] vgl. [Cran04]

[175] s.a. [Mein06]. Die Audit-Aufzeichnungen sind Kernbestandteil des „History Management“ zur Kontrolle der stattgefundenen Weitergaben eigener Daten.

3.5.5 Diskussion

Privacy Management Systeme sind ein Schritt in die Richtung der Vergabe von „Datenlizenzen“[176], da es hier möglich ist, die Daten verschlüsselt abzufragen, weiterzugeben und erneut zu speichern, ohne den Inhalt aufzudecken. Erst der Verarbeiter, der die Maßgaben der Policies erfüllt, erhält den Zugriff auf die unverschlüsselten Daten. Dies ist möglich, da die Policy-Überprüfung im Gegensatz zu Hippokratischen Datenbanken unabhängig vom Speicherort der Daten stattfindet. Durch das client-seitige PMS-Modul wird der Datenlebenszyklus schon früh begleitet (siehe Abbildung 12), die Policies entstehen durch Verhandlung zwischen Betroffenem und Verarbeiter.

Privacy Management Systeme

Policy-Management

Formulierungsort der Policies	Verhandlung zwischen Eigner und Verarbeiter
Perspektive der Policy-Formulierung	Eigner-Präferenzen, int. Policies des Verarbeiters
Management von Änderungen	Migration auf die neue Policy
Sprachen zur Formulierung	P3P (Eigner), EPAL (Verarbeiter)
Bindung an die Daten	„Datenlizenzen“

Durchsetzung der Policies

Zeitpunkt / Ort	Beim Betroffenen	Beim Verarbeiter	Bei Dritten
Präventiv, vor Übermittlung			
Präventiv, vor Verarbeitung		●	●
Detektivisch, nach Verarbeitung		●	
Nur Willenserklärung			

Wirkung im Datenlebenszyklus

Erstellen → Setzen in den Kontext → Verschlüsseln → Verteilen → Nutzen → Aktualisieren → Löschen → Dokumentieren

Abbildung 12: Steckbrief Privacy Management Systeme

3.6 Entwicklungspfad der Verfahren

Oben wurde gezeigt, dass die Verfahren, die einem umfassenden Datenschutz theoretisch sehr nahe kommen – die exklusive Speicherung beim Betroffenen einerseits und ein Privacy-DRM andererseits – an der Umsetzbarkeit unter den bestehenden Bedingungen scheitern. Als Vertreter der wichtigsten Verfahrensklassen zur Verarbeitungskontrolle wurden die klassische Access Control, Hippokratische Datenbanken (HippoDB) und

[176] vgl. [ChJo03]. Datenlizenzen gleichen Sticky Policies (vgl. Kapitel 2.3.3), indem sie den Daten anhaften, für die sie Gültigkeit besitzen. Dabei steht bei den Datenlizenzen eher die Zustimmungserklärung des ursprünglichen Dateneigners zu bestimmten Nutzungen im Vordergrund, als die Aufdeckung der Daten selbst.

Privacy Management Systeme (PMS) beschrieben. Deren Entwicklung über die Zeit und die jeweils wichtigsten Neuerungen zeigt Abbildung 13.

Bei der Grundform der Access Control werden Zugriffsrechte aufgrund der Kategorisierung der betroffenen Objekte an die Benutzer vergeben. Diese Kategorisierung und die Pflege der darauf basierenden Rechte obliegen dabei dem Betreiber des Datenspeichers. Dem wiederum folgt, dass der Zugriff von Administratoren des Betreibers nicht der Zugriffskontrolle unterliegt oder zumindest von diesen umgangen werden kann. Betroffene als Dateneigner haben keinen Einfluss auf die als Zugriffsrechte hinterlegten Policies. Und schließlich berücksichtigt Access Control in der Grundform weder den Kontext der Datenverarbeitung noch deren Zweck.

Dem gegenüber verfügen Hippokratische Datenbanken über die Möglichkeiten, Zweckbindung zu forcieren. Die Daten liegen in verschlüsselten Datenbanken und sind dort direkt mit Nutzungspolicies versehen (Sticky Policies). Zu Abfragen werden basierend auf dem angegebenen Zweck und den Rechten des Verarbeiters nur die legitimierten Datensätze und Attribute geliefert. Die Policies können im Gegensatz zur herkömmlichen Access Control aus den spezifischen Präferenzen der Betroffenen gebildet werden, aber auch hier liegt die Kontrolle über die Policies und deren Einhaltung wieder beim Betreiber der Datenbank und ist somit von dessen Zuverlässigkeit abhängig. Die Kontrollmöglichkeiten des Betroffenen sind limitiert, sofern überhaupt vorhanden. Ein Vorteil Hippokratischer Datenbanken ist neben der Zweckbindung die Abbildbarkeit von Verpflichtungen. Im Rahmen des Data Retention Management lassen sich etwa vertragliche oder gesetzliche Löschverpflichtungen verbindlich implementieren. Nachteilig auf die Verbreitung Hippokratischer Datenbanken wirkt dagegen die Notwendigkeit, spezielle Speichertechnologie bereit zu halten. Es ist nicht ohne Weiteres möglich, konventionelle Datenbankmanagementsysteme umzuwidmen.

Bei Privacy Management Systemen werden die zu schützenden Datenobjekte individuell verschlüsselt in den Datenspeichern abgelegt. Verarbeiter können dort Datensätze abrufen, speichern oder weitergeben, das heißt, im Gegensatz zu Hippokratischen Datenbanken erlaubt diese Technik die Delegation von Datenabfragen und die Distribution der verschlüsselten Inhalte. Aber es erhalten nur solche Verarbeiter die notwendigen Schlüssel zur Dechiffrierung aus dem Privacy Management System, die im richtigen Kontext die passenden Credentials und Zweckangaben liefern. PMS-Verfahren können mit herkömmlichen Datenbanken abgebildet werden, jedoch wird ein eigener PMS Server benötigt, der die Herausgabe der Schlüssel regelt. Diese Trennung bewirkt, dass Policies flexibler gehandhabt, also auch einfacher durch den Betroffenen beeinflusst und geändert werden können. Wie Access Control und die Hippokratischen Datenbanken entfalten auch PMS ihre datenschutzfördernde Wirkung erst nachdem die Daten initial gesammelt, dann mit Policies versehen und im Datenspeicher abgelegt sind. Die Schnittstelle von der Datenabgabe durch den Betroffenen bis zur Speicherung zur weiteren Nutzung bleibt ungeschützt. Auch teilen alle drei Verfahrensarten den Nachteil, dass Datenbank- bzw. Serverbetreiber die Policies lesen und daraus möglicherweise schon missbräuchlichen Informationsgewinn ziehen können.

Die beiden letztgenannten Nachteile sowie die oben genannte „Administratorenlücke“ zu vermeiden, sowie im Gegensatz zur datenschützerischen Nutzung von DRM-Techniken oder der Speicherung beim Betroffenen realistisch umsetzbar zu sein, sollte folglich Ziel von Verfahren sein, die einen Fortschritt im Sinne des Datenschutzes erzielen wollen.

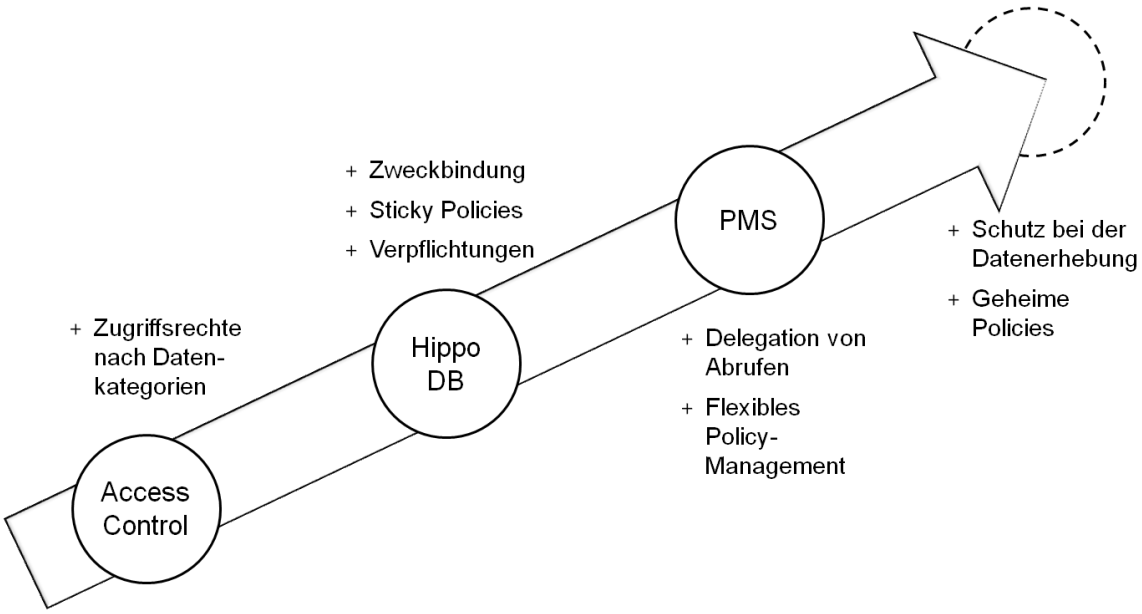


Abbildung 13: Entwicklungspfad der Verfahren zur Verarbeitungskontrolle

3.7 Schlussfolgerungen

Vorstehend hat sich gezeigt, dass ein Privacy-DRM wenig Chance auf Realisierung hat. Würde dies dennoch gelingen, hätte es zudem größere Probleme mit der analogen Schnittstelle als es die DRM-Systeme mit Medien-Fokus haben. Im Gegensatz zu Multimedia-Daten sind sensible personenbezogene Daten oft mit nur wenigen Kilobytes zu speichern, im Gesamtzusammenhang schon auf einer Handvoll Bildschirmseiten darstellbar und vielfach für einen aufmerksamen Beobachter nach einmaliger Ansicht gut einzuprägen. Das Risiko, dass ein Verarbeiter nach einer erstmaligen Aufdeckung die Daten also druckt, kopiert, abschreibt, durch Bildschirmabgriffe sichert oder sich schlicht die Kernfakten merkt, ist signifikant.

Sucht man neben DRM-Systemen andere Optionen zur Kontrolle des Lebenszyklus herausgegebener Daten, stößt man auf Lösungen wie Vanish[177] oder xpire![178], die jedoch über den Charakter von punktuell begrenzten Lösungen nicht hinauskommen und sich ebenfalls mindestens der Problematik der analogen Weiterverarbeitung geschlagen geben müssen.

Fasst man die vorgenannten Überlegungen zu einem Privacy-DRM und dem Schutz personenbezogener Daten über ihren gesamten Lebenszyklus zusammen, muss man schließen, dass keine Lösung bekannt ist, die Daten vor künftigem Missbrauch technisch wirksam schützt, wenn diese bereits einem Verarbeiter im Klartext vorliegen. Ihr Schutz kann nur noch nachgelagert, mit juristischen Mitteln, erfolgen.

Im Umkehrschluss muss sich technischer Datenschutz besonders auf das Machbare fokussieren, nämlich den möglichst weitgehenden Schutz der Daten vor ihrer ersten Kenntnisnahme durch den Verarbeiter. Dabei ist zu gewährleisten, dass der Verarbeiter bei Vorliegen vereinbarter Sachverhalte zuverlässig an die Daten gelangt. Ein gutes System ist also ein System, das personenbezogene Daten exakt bis zum Zeitpunkt der konkret veranlassten Nutzung geheim hält.

[177] vgl. [GKLL09]

[178] vgl. [BBDG+11]. Backes et al. spezialisieren ihre Lösung darauf, Bilder, die in sozialen Netzwerken wie Facebook hochgeladen werden, mit einem Datum zu versehen, an dem der für die Darstellung der Bilder auf einem Schlüsselserver abgelegte Schlüssel ungültig wird und damit das Bild nicht mehr dargestellt werden kann. Federrath et al. zeigen, wie leicht sich die Schwachstellen selbst einer so spezialisierten Lösung nutzen lassen, um deren Sicherheit auszuhebeln [FFHM+11].

4 Motivation eines neuen Verfahrens

4.1 Untersuchung der Rahmenbedingungen

Bei Betrachtung der anfangs genannten Beispiele zu Datenbereitstellungen der Kategorie „Applikation“ lassen sich die folgenden Gemeinsamkeiten erkennen:

- Ziel und Zweck der Verarbeitung stehen für den Betroffenen fest.
- Der Zeitpunkt der potentiellen Verarbeitung ist jedoch möglicherweise unbekannt.
- Die Population der gewünschten Verarbeiter ist dynamisch und nur unvollständig bekannt.

Der letztgenannte Punkt lässt sich gut nachvollziehen, wenn man die Population der gewünschten Verarbeiter etwa auf die Gruppe aller Ärzte eines Bundeslandes, die deutschen Strafverfolgungsbehörden oder eine Liste aller Kunden oder Lieferanten einer Organisation bezieht. Der Aufwand einer individuellen Pflege solcher Listen wäre erheblich, teilweise in Ermangelung geeigneter öffentlicher Informationsquellen für Privatpersonen sogar unmöglich.

Betroffene können also ihre eigenen Präferenzen formulieren und aktuell halten. Dass sie jedoch die Dynamik des von ihnen angesprochenen Umfelds konsequent in Eigeninitiative abbilden können, darf nicht in allen Fällen erwartet werden.

Es ergeben sich die ersten Forderungen an ein zu formulierendes Verfahren:

Policy-Formulierung

Vom Betroffenen formulierte Policies sind in höherem Maße zur Wahrnehmung seiner datenschützerischen Interessen geeignet als von irgendeiner anderen Partei formulierte Policies. In den Policies muss er mindestens festlegen können, zu welchem Zweck welche seiner Daten an welchen Verarbeiter herausgegeben werden.

Asynchrones Policy-Enforcement

Die Einhaltung der formulierten Policies soll unabhängig von der momentanen Verfügbarkeit des Betroffenen zum Zeitpunkt der entsprechenden Verarbeiter-Anfragen durchgesetzt werden. Das bedeutet im Umkehrschluss, dass die Policies vom Betroffenen zu einem beliebigen Zeitpunkt vor der Datenbereitstellung formuliert werden können.

Variable Empfänger

Verarbeiter sind dem Betroffenen entweder individuell bekannt oder er kann Eigenschaften beschreiben, die einen Verarbeiter als Mitglied einer Gruppe berechtigter Datenempfänger ausweisen können. Diese Gruppen müssen Veränderungen über die Zeit abbilden.

Will man datenschutzfreundliche Technik gegenüber Verarbeitern durchsetzen, bieten sich zwei Strategien an: Zum einen die juristische Verpflichtung, zum anderen der kommerzielle Anreiz. Rechtliche Anordnung findet ihre Grenzen in der länder- und kontinentübergreifenden Natur der Datenverarbeitungssysteme. Also muss ein Anreizsystem bestehen, wenn man Verarbeiter davon überzeugen will, in entsprechende Maßnahmen zu investieren.

Garantierte Daten-Lieferung

Ein Anreiz ist die garantierte Daten-Lieferung. Kann ein Verarbeiter davon ausgehen, dass die vereinbarten Daten bereitstehen, wenn der die Nutzung zweckbegründende Sachverhalt eintritt, reduziert er das Risiko eines Vertragsbruchs durch sein Gegenüber und die damit möglicherweise verbundenen finanziellen Verluste. Dies geschieht zum einen dadurch, dass die Übermittlung der Klartextdaten unabhängig davon erfolgen kann, ob der Dateneigner zum betreffenden Zeitpunkt verfügbar und entscheidungsfähig ist. Zum anderen wird die garantierte Datenlieferung dadurch abgesichert, dass ein Verarbeiter jederzeit vor Eintreten des Nutzungszwecks prüfen kann, ob Daten bereitgestellt wurden – ohne eine Einsicht in dieselben zu gewinnen.

Nachweisbarkeit

Weiterhin dem Schutz des Verarbeiters dient die Forderung, dass die Datenübermittlung und deren Quelle, der Dateneigner, im Streitfall nachgewiesen werden können. Könnte ein Dateneigner erfolgreich leugnen, ein Datum bereitgestellt zu haben, wäre die Verwertbarkeit für den Verarbeiter deutlich eingeschränkt oder würde gar zu seinem eigenen juristischen Risiko werden. Auch für den Betroffenen ist der Nachweis einer erfolgten Datenübermittlung an den Verarbeiter von Interesse. Nur damit hat er die grundlegenden Mittel an der Hand, einen möglichen Missbrauch seiner personenbezogenen Daten zu verfolgen[179].

Sicherheit

Vertraulichkeit und Integrität der Daten vervollständigen den Katalog der Anforderungen an ein System, das personenbezogenen Daten der Kategorie „Applikation“ in den genannten Parametern schützen kann. Außer dem Betroffenen und dem Verarbeiter – nach vorliegen der Voraussetzungen – soll keine weitere Partei Einblick in die Daten, die Art der Daten oder die Policies erhalten. Ein unerkanntes Verfälschen oder Austauschen der Datensätze soll ebenfalls verhindert werden.

Dabei spielt die Sicherung der Infrastruktur, also der Verarbeitungssysteme und Netze, gegen fremde Eingriffe eine Rolle. Zuverlässiger ist jedoch die unmittelbare daten-zentrische Sicherheit, die durch Verschlüsselung der Daten mit angemessener Kryptographie erreicht wird. Dieses stärkt ein System gegenüber Schwachstellen in den verwendeten Datenbanken, Betriebssystemen und Übertragungsprotokollen.

[179] Aus der juristischen Diskussion stammen die vier Ziele datenschützerischen Handelns, Verfügbarkeit, Integrität, Vertraulichkeit und Verbindlichkeit (s. etwa [GoKl03]). In diesem Sinne entspricht die Garantierte Daten-Lieferung der Verfügbarkeit und die Nachweisbarkeit korrespondiert mit der Verbindlichkeit.

Fasst man die formulierten Anforderungen zusammen, wird ein System gesucht, das personenbezogene Daten bis zum Zeitpunkt der konkret veranlassten Nutzung geheim hält, zuverlässig liefern kann und die Präferenzen des Betroffenen dynamisch implementiert.

4.2 Anforderungen

Konkret vorgeschlagen wird ein Verfahren mit den folgenden Zielen:

I Der Betroffene kann zu einem beliebigen Zeitpunkt vor Eintreten eines entsprechenden Anlasses festlegen, zu welchem Zweck welche Daten für welchen Verarbeiter verfügbar sein sollen.

II Dabei muss der Verarbeiter zum Zeitpunkt der Festlegung nicht unbedingt individuell bekannt sein. Der Betroffene kann die Freigabe seiner Daten auch einer variablen Gruppe, deren Mitglieder er nicht dediziert kennt, zugestehen. (Dieser Aspekt wird in Szenario 2 dieser Arbeit realisiert.)

III Der Erfolg einer Datenanfrage und -übermittlung ist nicht davon abhängig, ob ein Betroffener zum entsprechenden Zeitpunkt online ist. Dies ist beispielsweise im Fall von medizinischen Notfällen essentiell.

IV Ein Verarbeiter kann vorab prüfen, ob er im Bedarfsfall davon ausgehen kann, Daten zu erhalten, d.h. ob Daten für eine bestimmten Anlass bereit stehen. Sollte diese Bereitstellung zu einem späteren Zeitpunkt vom Betroffenen zurückgenommen werden, wird der Verarbeiter hierüber informiert, sofern er die Daten nicht bereits abgerufen hat.

V Eine erfolgte Übermittlung von Daten eines Betroffenen kann später von keiner der beteiligten Parteien geleugnet werden.

VI Der Betroffene kann jederzeit in Erfahrung bringen, an wen welche seiner Datensätze mit welcher Zweckangabe übermittelt wurden.

VII Außer dem Betroffenen und im Übermittlungsfall dem Verarbeiter gibt es keine weiteren Parteien, die Einsicht in die Daten, die Art der Daten oder die hinterlegten Verwendungszwecke haben. Idealerweise soll auch niemand Rückschlüsse aus der Menge der hinterlegten Daten oder der Häufigkeit ihrer Abfrage ziehen können. Niemandem außer dem Dateneigner und dem Verarbeiter sollen Interaktionen der beiden erkennbar sein.

4.3 Beispielhafte Anwendungsfälle

Ein Verfahren, das die oben formulierten Anforderungen erfüllt, eignet sich unter anderem für die folgenden Einsatzbeispiele.

4.3.1 Ärztliche Betreuung

Ein Angestellter verpflichtet sich bei seiner Einstellung in einem Unternehmen, bestimmte Informationen für den Betriebsarzt zu hinterlegen, die diesem im Falle eines Betriebsunfalls oder plötzlicher schwerer Erkrankung zur Verfügung stehen sollen. Um im Notfall die richtigen Maßnahmen ergreifen zu können, benötigt der Betriebsarzt beispielsweise Daten zur Blutgruppe des Betroffenen, zu Vorerkrankungen, regelmäßig eingenommenen Medikamenten, Allergien oder chronischen Krankheiten. Unabhängig von der Schweigepflicht des Arztes will der Angestellte aus nachvollziehbaren Gründen nicht, dass ohne konkretes Vorliegen eines Notfalls dem Betriebsarzt und seinen Mitarbeitern zur Kenntnis gelangt, dass er etwa unter Bluthochdruck oder chronischen Bandscheibenproblemen leidet. Damit der Betriebsarzt aber seinem Auftrag nachkommen kann, verlangt er eine Zusicherung, dass er im Notfall Zugriff auf die relevanten Daten erlangt, diese also an einer ihm zugänglichen Stelle hinterlegt sind und er ohne Zeitverzug und ohne unmittelbare Mitwirkung des Betroffenen diese Daten zur Erstversorgung nutzen kann.

Analog gelagert sind auch Fälle, in denen ein im Notfall erstversorgender Arzt nicht wie ein Betriebsarzt vorab bekannt ist, sondern zunächst nur über die Zugehörigkeit zur Gruppe „Notfall-Ärzte“ identifiziert wird. Ohne dass der konkret an einem Unfallort eintreffende Arzt vorab den Namen des Betroffenen kennt, soll er dessen medizinisch relevante Daten unverzüglich erlangen können, sobald zumindest die Identität des zu Versorgenden ermittelt ist. Hierzu hat sich der Notfall-Arzt bereits vor seinen Einsätzen bei einer Zertifizierungsinstanz in die Gruppe „Notfall-Ärzte“ eingeschrieben. Die Zertifizierungsstelle hat die Eigenschaft als Notfall-Arzt nach Vorlage der erforderlichen Nachweise bestätigt.

Der Betroffene hat wiederum durch Teilnahme am Verfahren zugestimmt, dem Urteil der Zertifizierungsinstanz zu vertrauen.

4.3.2 Verdachtsbegründeter Systemaudit

Ein Unternehmen verpflichtet sich – etwa unter Auflagen eines Gerichts oder der Börsenaufsicht – im Falle eines konkreten Verdachtsmoments finanzieller Manipulationen externen Auditoren Zugriff auf Transaktionsdaten von Mitarbeitern zu gewähren, die mit der IT-seitigen Steuerung von Zahlungsflüssen betraut sind. In turnusmäßigen verdachtsunabhängigen Prüfungen werden die Realnamen der Mitarbeiter ausgeblendet,

um die Möglichkeiten einer unzulässigen Arbeitskontrolle durch den Arbeitgeber auszuschließen. Zur Prüfung eines konkreten Verdachts wird jedoch die Auswertung auch der Benutzerdaten als notwendig angesehen und durch den Betriebsrat akzeptiert.

Mitglieder einer zertifizierten Gruppe „Auditoren“ sollen also ohne vorherige Inkenntnissetzung des geprüften Unternehmens die Möglichkeit erhalten, in der Systemanalyse gefundene Benutzer-IDs den echten Namen der Anwender zuordnen zu können. Eine entsprechende Zuordnungsliste soll vertraulich hinterlegt und nach Angabe des Verwendungszwecks „Verdachtsbegründeter System-Audit“ für die Auditoren abrufbar sein. Das Unternehmen ist im Nachgang von der erfolgten Abfrage der Liste in Kenntnis zu setzen.

4.3.3 Geschäftsreisen

Geschäftsreisende, die ins Ausland reisen, um Geschäftspartner zu treffen, wollen aus Furcht vor Industriespionage ihren Aufenthaltsort und die besuchten Personen vertraulich halten. Allerdings müssen Kontaktinformationen bereitliegen, falls das Auswärtige Amt eine Reisewarnung herausgibt oder ein Konsulat den Reisenden im Falle eines heimischen Notfalls zurückrufen muss.

Auch erzeugen Computerreservierungssysteme für Reisen umfangreiche Passagiernamensregister (Passenger Name Records – PNR) mit personenbezogenen Daten ihrer Reisegäste[180]. Einige Länder fordern die Hinterlegung dieser Register, bevor sie eine Einreise der betreffenden Personen zulassen. Die Teile des PNR, die nur im Falle eines konkreten kriminalistischen Verdachts oder einer Notsituation benötigt werden, könnten mithilfe eines entsprechenden Verfahrens vor dem Eintreten dieser Notwendigkeit geschützt werden.

4.3.4 Anonyme Geschäfte

Auf einem Online Marktplatz bzw. einer Auktionsplattform für elektronische Güter (Musik, Videos) treten Käufer nur mit ihren Nicknames auf. Kommt es zum Abschluss, werden die finanzielle Transaktion und die Übermittlung der Inhalte durch einen Treuhänder-Service des Plattform-Betreibers abgewickelt, wobei Käufer und Verkäufer voreinander anonym bleiben. Kommt es jedoch zu einer Beschwerde seitens des Käufers, weil die Inhalte nicht die zugesagten Eigenschaften aufweisen, erwartet er eine Garantie, dass er die echten Kontaktdaten des Verkäufers in Erfahrung bringen kann[181].

[180] vgl. [Fox10]

[181] Hier nimmt der Plattform-Betreiber die Rolle des „Identitäts-Treuhänders“ ein, wie etwa auch in [FePf03] beschrieben.

4.3.5 Selektive Abwesenheitsnotiz

Der Geschäftsführer eines Unternehmens möchte seinen Bestandskunden oder Unternehmen einer bestimmten Branche eine konkrete Abwesenheitsnotiz zukommen lassen, sofern sich diese während eines längeren Urlaubs an ihn wenden[182]. Anderen Unternehmen, unter denen sich seine Konkurrenz befinden könnte, möchte er jedoch keinen Hinweis auf seine Abwesenheit geben.

4.3.6 Reaktivierung

Verlässt ein Mitarbeiter sein Unternehmen in den vorläufigen Ruhestand, ist dieses nach entsprechenden Löschfristen dazu verpflichtet, die personenbezogenen Daten des Mitarbeiters zu löschen. Es kann jedoch sinnvoll sein, die Daten für den Fall einer späteren Reaktivierung des Mitarbeiters oder anfallender Nachfragen zum Zweck einer Rentenneuberechnung wieder in den Zugriff zu bekommen. Das Unternehmen könnte dem Mitarbeiter zum Zeitpunkt der Löschung eine signierte Sammlung seiner Daten übergeben und mit ihm vereinbaren, dass er sie an einem sicheren Ort so ablegt, dass das Unternehmen in den definierten Fällen wieder Zugriff auf diese erlangen kann.

[182] vgl. [Gerh09] S.85f, zur datenschützerischen Relevanz von generierten Abwesenheitsnotizen.

5 Vorstellung des Verfahrens PDG

Mit dem „Purpose sensitive data provisioning guard“[183] (PDG) wird in den folgenden Abschnitten ein neues Verfahren vorgeschlagen und untersucht, das die oben formulierten Anforderungen zum Ziel nimmt.

5.1 Einführung der Szenarien

Das Vorgehen ist zweistufig. Zunächst erfolgt die Vorstellung eines vereinfachten Schemas (Szenario 1: PDG), das die grundsätzlichen Überlegungen und Vorgehensweisen verdeutlicht. Das anschließend beschriebene Szenario 2: PDG/v (PDG with variable processor groups)[184] bildet dann das vollständige Protokoll ab und hebt die zur Vereinfachung in Szenario 1 eingeführten Beschränkungen auf.

5.1.1 Szenario 1: Bekannte Empfänger

Im Gegensatz zum später zu beschreibenden Szenario 2 weiß der Dateneigner hier, wem er seine personenbezogenen Daten in definierten Situationen zukommen lassen will. Jeder potentielle Verarbeiter seiner Daten ist ihm also bekannt. So kann er Regeln (Policies) darüber definieren, wer den Inhalt eines bestimmten Datentyps künftig einsehen darf, wenn dieser einen hierfür geeigneten Verwendungszweck angibt.

Der Verarbeiter „Betriebsarzt Dr. Müller“ soll beispielsweise Zugriff auf den Datensatz vom Typ „Krankengeschichte“ erhalten, wenn der Zweck „Arbeitsunfall“ vorliegt. Für den Zweck „Statistische Erhebung“ würde der Dateneigner hingegen dem Betriebsarzt keinen Zugriff erlauben. Ebenso wenig würde er den Zugriff auf diese Daten durch den Verarbeiter „Reisebüro International“ gestatten, egal zu welchem Zweck dieses angefragt wird. Für das Reisebüro möchte er allerdings die Daten zu seinen jeweils aktuellen Schutzimpfungen vorhalten, gebunden an den Verarbeitungszweck „Reisewarnung“.

Policies definieren also pro Dateneigner die verschiedenen Tripel aus „Datentyp“, „Verarbeitungszweck“ und „Empfänger“.

Die Daten sollen dem jeweiligen Verarbeiter nicht auf Verdacht, sondern erst bei Vorliegen eines rechtfertigenden Grundes, also Verarbeitungszwecks, zugänglich gemacht werden. Es ist vorab nicht bekannt, ob und wann dieser Fall eintreten wird. Zugleich kann der Dateneigener nicht sicherstellen, dass er zum fraglichen Zeitpunkt erreichbar und auskunftsfähig sein wird. Daher hinterlegt er die Daten, gebunden an die zugehörigen Policies, bei einer zentralen Aufbewahrungsstelle. Sowohl die Daten als auch die Policies

[183] Dt.: „Wächter über die zweckgebundene Bereitstellung von Daten“.
[184] Dt.: „PDG mit variablen Empfängergruppen“.

sind dabei so verschlüsselt, dass die Aufbewahrungsstelle keine Kenntnis der Inhalte erlangen kann. Auch die Identität des Dateneigners wird ihr nicht bekannt gemacht. Erst der vorgesehene Verarbeiter kann unter Angabe des geeigneten Verarbeitungszwecks die Daten an der Aufbewahrungsstelle abrufen (fortan als „Inhaltsabfrage“ bezeichnet) und entschlüsseln. Dies gelingt natürlich nur, wenn dediziert für ihn und den angegebenen Verwendungszweck der angeforderte Datentyp hinterlegt ist.

Selbst wenn die konkrete Notwendigkeit zur Datenverarbeitung noch nicht vorliegt, ist es gegebenenfalls für den künftigen Verarbeiter vorab wichtig zu wissen, ob ein bestimmter Datensatz für ihn vorliegt. Hierfür kann er im Rahmen einer „Existenzanfrage“ an die Aufbewahrungsstelle die entsprechende Policy übermitteln. Er stellt also die Frage: „Gibt es von dem Dateneigner für mich einen Datensatz von einem bestimmten Typ, den ich unter Angabe des Verwendungszwecks abrufen könnte?“ Der Aufbewahrer übermittelt in diesem Fall als Rückmeldung nicht den verschlüsselten Datensatz, sondern nur die Information, ob der Datensatz bei ihm hinterlegt ist[185]. Dabei ist die angefragte Policy durch den Verarbeiter mit demselben Schlüssel verschlüsselt wie die gespeicherte Policy. Die Aufbewahrungsstelle führt eine Vergleichsoperation zwischen für sie bedeutungslosen Zeichenfolgen (den verschlüsselten Policies) durch und kann bei einer Übereinstimmung die entsprechende Rückmeldung geben, ohne selbst zu wissen, welche Policy abgefragt wurde. Auf diese Weise kann etwa der Betriebsarzt prüfen, ob er im Notfall die benötigten Daten zur Krankengeschichte eines bestimmten Patienten vorliegen hat. Dieses könnte beispielsweise eine vertragliche Bedingung des Arbeitsverhältnisses sein.

Um Missbrauch durch Verarbeiter sanktionieren zu können, protokolliert die Verarbeitungsstelle jede Existenz- und Inhaltsabfrage. Sie stellt dem Dateneigner die Abrufprotokolle zu dessen Überprüfung zur Verfügung.

Personenbezogene Daten sind im Zeitablauf Änderungen unterworfen, und so kann der Dateneigner die bei der Aufbewahrungsstelle hinterlegten Datensätze erweitern, modifizieren sowie Sätze oder Policies entfernen. Falls für eine jetzt modifizierte Policy in der Vergangenheit bereits erfolgreiche Existenzanfragen stattgefunden haben, führt die Aufbewahrungsstelle die entsprechende Vergleichsprüfung nun ein weiteres Mal durch und informiert den Verarbeiter, falls keine Übereinstimmung mehr vorliegt. Anderenfalls würden die Verarbeiter auf die Ergebnisse veralteter Existenzanfragen vertrauen, obwohl möglicherweise die benötigten Daten nicht mehr für den Abruf vorgehalten werden.[186]

Die Aufbewahrungsstelle kann neben der beschriebenen Tätigkeit auch weitere Dienste anbieten. Sie könnte beispielsweise Zuverlässigkeitsrankings über Verarbeiter führen oder

[185] Da der Betroffene den Datensatz explizit für den Verarbeiter hinterlegt, akzeptiert er die Modalitäten des Protokolls und willigt in die Nutzung von Existenzanfragen ein. Sollten in einzelnen Nutzungskontexten jedoch die Antworten auf Existenzanfragen bereits ein datenschutzrechtliches Problem darstellen, kann die Möglichkeit der Existenzanfrage ohne weitere Einflüsse auf die sonstige Funktionalität unterbunden werden.

[186] Es ist ersichtlich, dass diese Aktualisierungen bei sehr dynamischen Datenstrukturen, die in kurzen Intervallen entstehen und vergehen, aufwendig werden. Im geschilderten Nutzungskontext des Protokolls, der Ablage vom Betroffenen formulierter personenbezogener Daten, sind diese Umschlagsraten jedoch nicht zu erwarten.

Datenübermittlungen an Verarbeiter sperren, die sich verdächtig verhalten. Ebenso könnte die Aufbewahrungsstelle Verfallsdaten zu den einzelnen Paketen verwalten. Die letztgenannten Punkte werden im Folgenden nicht weiter verfolgt, da sie keinen signifikanten Einfluss auf die Ausgestaltung der hier betrachteten Protokolle haben.

5.1.2 Szenario 2: Variable Empfängergruppen

Szenario 1 folgt der Prämisse, dass der Dateneigner alle potentiellen Verarbeiter kennt, und diese zu dem Zeitpunkt benennen kann, zu dem er seine Daten der Aufbewahrungsstelle übergibt.

Die Prämisse kann aus verschiedenen Gründen zu kurz greifen:

- Möglicherweise ist die Gruppe der möglichen Verarbeiter, denen Daten zur Verfügung zu stellen sind, sehr groß. Dies könnte die Erfassung und fortlaufende Pflege durch den Dateneigner zu einem unrealistisch aufwendigen Unterfangen machen. Möchte er zum Beispiel einen Datensatz für alle Kollegen seines Unternehmens vorhalten, kann dies bei entsprechender Unternehmensgröße mühsam sein.
- Weiterhin kann es sein, dass eine stetige Erweiterung der Liste potentieller Empfänger eines Datensatzes absehbar ist, mit der ein Dateneigner nicht Schritt halten kann, etwa die Mitglieder eines neu gegründeten Vereins.
- Im dritten Fall sind dem Dateneigner die potentiellen künftigen Verarbeiter seiner Daten generell nicht namentlich bekannt. Er kann sie zwar einer Gruppe zuordnen („Notfallärzte“, „Strafverfolgungsbehörden“...), aber die einzelnen Mitglieder nicht identifizieren.

Das Vorgehen nach Szenario 1 kann in diesen Fällen nicht angewandt werden. Da dem Dateneigner die künftigen Empfänger nicht bekannt sind, kann er die Datensätze vor Hinterlegung bei der Aufbewahrungsstelle nicht in der Form bereitstellen, dass sie direkt von den Empfängern abgerufen und entschlüsselt werden können.

Dennoch kann er Policies für seine Daten erstellen. Diese bestehen jedoch nicht aus den Tripeln von „Datentyp“, „Verarbeitungszweck“ und „Empfänger“, sondern „Datentyp“, „Verarbeitungszweck“ und generischen „Empfängergruppen“. Letztere bezeichnen eine Gruppe von potentiellen Datenverarbeitenden, die unter einer gemeinsamen Kombination von Datentyp und Verarbeitungszweck auf einen Datensatz zugreifen dürfen. Einer Empfängergruppe „Notfallärzte“ würde man etwa Zugriff auf die eigene Krankengeschichte unter der Bindung an den Zweck „medizinischer Notfall“ geben.

Die veränderte Situation bedarf einiger Anpassungen gegenüber Szenario 1. Zum einen wird eine Instanz benötigt, die prüft, ob Datenverarbeiter den Zugang zu einer der generischen Empfängergruppen erhalten. Sie lässt sich beispielsweise den Nachweis erbringen, dass ein neuer Datenverarbeiter, der Zutritt zur Gruppe „Notfallärzte“ beantragt,

Mediziner ist und darüber hinaus die formellen Zulassungskriterien zur Notfallversorgung erfüllt. Die Instanz „zertifiziert“ also potentielle Kandidaten für die Teilnahme an dem Szenario. Im Folgenden wird sie daher als Zertifizierungsinstanz (oder kurz Zertifizierer) bezeichnet. Der Zertifizierer ist weder identisch mit dem Dateneigner – dies würde den Voraussetzungen des Szenario 1 entsprechen –, noch kann diese Rolle an die Aufbewahrungsinstanz vergeben werden. Sie darf auch, über die notwendigen Schritte des Protokolls hinaus, nicht mit dem Aufbewahrer kooperieren. Diese beiden Parteien könnten anderenfalls durch Kombination der bei ihnen jeweils hinterlegten Daten Rückschlüsse auf die sensitiven personenbezogenen Daten der Betroffenen ziehen. Diese Problematik wird in der Diskussion der Angreifermodelle detailliert aufgegriffen.

Zum anderen besteht die Herausforderung, dass dem Dateneigner zu dem Zeitpunkt, an dem er seine Daten bei der Aufbewahrungsstelle hinterlegt, nicht die öffentlichen kryptographischen Schlüssel der künftigen Verarbeiter bekannt sind. Er kennt schließlich die Verarbeiter selbst noch nicht. Daher muss die Zertifizierungsstelle neben der Verwaltung der Empfängergruppen eine zweite Aufgabe wahrnehmen. Sie dient allen Dateneignern als virtueller Empfänger Ihrer Daten, das heißt, Dateneigner verschlüsseln Ihre Daten mit empfängergruppenspezifischen Schlüsseln und legen diese beim Zertifizierer ab. Unter Einbeziehen des Aufbewahrers wird dabei ein zeitversetzter Schlüsseltransfer von den Dateneignern zu Mitgliedern der Empfängergruppen realisiert, bei dem weder die Aufbewahrungs- noch die Zertifizierungsinstanz den Schlüssel und somit die konkreten Daten erfahren.

Datenverarbeiter, die durch den Zertifizierer bestätigte Mitglieder einer betreffenden Empfängergruppe sind, können durch ein Modell zum Schlüsselaustausch einen kryptographischen Schlüssel vom Dateneigner erhalten, mit dem sie künftige Existenz- und Inhaltsabfragen stellen können. An dem Schlüsselaustausch ist sowohl die Aufbewahrungsstelle als auch der Zertifizierer beteiligt. Beide erfahren dabei nicht, wie der Schlüssel im Klartext aussieht. Obwohl der Dateneigner zum Zeitpunkt des Schlüsselaustauschs nicht erreichbar sein muss und er den anfragenden Verarbeiter möglicherweise nicht kennt, gelingt also der protokollierte Transfer seines geheimen Schlüssels zu dem Verarbeiter. In der Folge kann dem Protokoll der Existenz- und Inhaltsabfragen zwischen Verarbeiter und Aufbewahrer weiter gefolgt werden.

5.2 Protokolle

Im Folgenden sei:

Betroffener / Dateneigner $i \in \{1, 2, \ldots, n\}$ mit n = Anzahl der Individuen, die ihre Daten hinterlegen	A_i
Datenverwender / -verarbeiter $j \in \{1, 2, \ldots, m\}$ mit m = Anzahl der Verarbeiter, die potentiell Daten der Individuen abrufen	B_j
Aufbewahrungsstelle der Daten	T
Zertifizierungsstelle [Szenario 2]	Z
Zweck der Datenverarbeitung $k \in \{1, 2, \ldots, o\}$ mit o = Anzahl der vereinbarten Verwendungszwecke	J_k
Typ eines Datensatzes (inhaltliche Qualifizierung) $l \in \{1, 2, \ldots, p\}$ mit p = Anzahl der vereinbarten Datensatztypen	Y_l
Satz personenbezogener Daten des Typs Y_l von und über den Betroffenen A_i	$M(A_i, Y_l) = M_{il}$
Symmetrischer Schlüssel (zwischen A_i und B_j)	$K(A_i, B_j) = K_{ij}$
Verschlüsselung von M mit symmetrischem Schlüssel K_{ij}	$E_{ij}(M) = C$
Entschlüsselung von C mit symmetrischem Schlüssel K_{ij}	$D_{ij}(C) = M$
Verschlüsselung von M mit öffentlichem Schlüssel von B_j	$E_j(M) = C$
Entschlüsseln von C mit privatem Schlüssel von B_j	$D_j(C) = M$
Policy, d.h. eine Kombination von Datensatztyp Y_l und Verarbeitungszweck J_k, verschlüsselt mit K_{ij}	$E_{ij}(Y_l, J_k) = P_{ijlk}$
Signatur von M mit dem privaten Signaturschlüssel von A_i	$S_i(M)$
Testen einer Signatur von A_i	$V_i(S_i(M))$
Öffentlicher Schlüssel der Partei x	$K_{pub\ x}$
Privater / geheimer Schlüssel der Partei x	$K_{priv\ x}$

Durch Dateneigner A_i generierte Zufallszahl $x \in \{1, 2, \ldots, z\}$ mit z = Anzahl der von A_i insgesamt generierten Datensätze	r_{ix}
Erzeugen eines Hashwerts durch Anwendung einer Einweg-Hashfunktion h auf M	h(M)

5.2.1 Schritte zur Einführung

Zunächst einigen sich Dateneigner und -verarbeiter, etwa über Interessenvertretungen oder Normierungsorganisationen, auf einen Katalog zulässiger Datentypen Y_l und Zweckdefinitionen J_k zur Verarbeitung personenbezogener Daten.

$J_k \in$ {„medizinischer Notfall“, „Steuerprüfung“, „Vereinsauflösung“, …}
$Y_l \in$ {„Krankengeschichte“, „Religionszugehörigkeit“, „Mitgliedschaft“…}

0.1 Die Aufbewahrungsstelle T akzeptiert und veröffentlicht den Katalog zulässiger Datentypen $Y_1, Y_2 \ldots Y_p$

0.2 T akzeptiert und veröffentlicht den Katalog von Zweckdefinitionen $J_1, J_2 \ldots J_o$

Eine exakte Festlegung und später buchstabengetreue und ausschließliche Verwendung der Elemente aus diesen Katalogen ist notwendig. Dateneigner werden bei Hinterlegung ihrer Datensätze diese Elemente nutzen, um Policies zu erstellen. Datenverarbeiter werden ihre Anfragen ebenfalls basierend auf den Elementen der Kataloge formulieren. Beide werden in späteren Schritten der Protokolle gemeinsame symmetrische Schlüssel darauf anwenden. Nur wenn beide so entstehenden Chiffrate Bit für Bit übereinstimmen, kann T bei Anfragen eines Verarbeiters eine Beziehung zwischen Anfrage und gespeichertem Datensatz herstellen. Er hat keine Einsicht in die konkreten Inhalte der Chiffrate und kann so nicht etwa eine näherungsweise Prüfung „ähnlicher“ Policies vornehmen.

Wer die Kataloge zu Datentypen und Verarbeitungszwecken veröffentlicht, ist im Grunde unerheblich. Sie müssen allen potentiellen Dateneignern und Verarbeitern bekannt sein und es besteht kein Grund, sie vor weiteren Parteien, wie etwa T geheim zu halten.

Die „Platform for Privacy Preferences“ (P3P) des W3C umfasst die Definition von 12 Standard-Verwendungszwecken[187]. Unter anderem wegen der häufig kritisierten Vermischung von Verarbeitungszielen und -arten in P3P[188] können diese jedoch nur als Ausgangspunkt für eine Nutzungsart-bezogene Definition der J_k dienen.

[187] vgl. [Greß01], [GaSp02]
[188] vgl. [Zieg02], [Feda07], [EPJu00]

5.2.2 Protokolle für Szenario 1

Abbildung 14 zeigt schematisch den Ablauf des Protokolls für Szenario 1. Die Schritte werden in den darauf folgenden Abschnitten erläutert, wobei die Nummerierung der in der Grafik verwendeten entspricht.

Die Abfrage der beim Aufbewahrer gespeicherten Protokolle durch den Dateneigner (Schritte 5.1 bis 5.4) sowie das Durchführen von Änderungen an den hinterlegten Daten (Schritte 6.1 bis 6.8) spielen für das Verständnis der Kernfunktionalität eine untergeordnete Rolle und sind in der Abbildung vereinfachend zu 5.x und 6.x zusammengefasst.

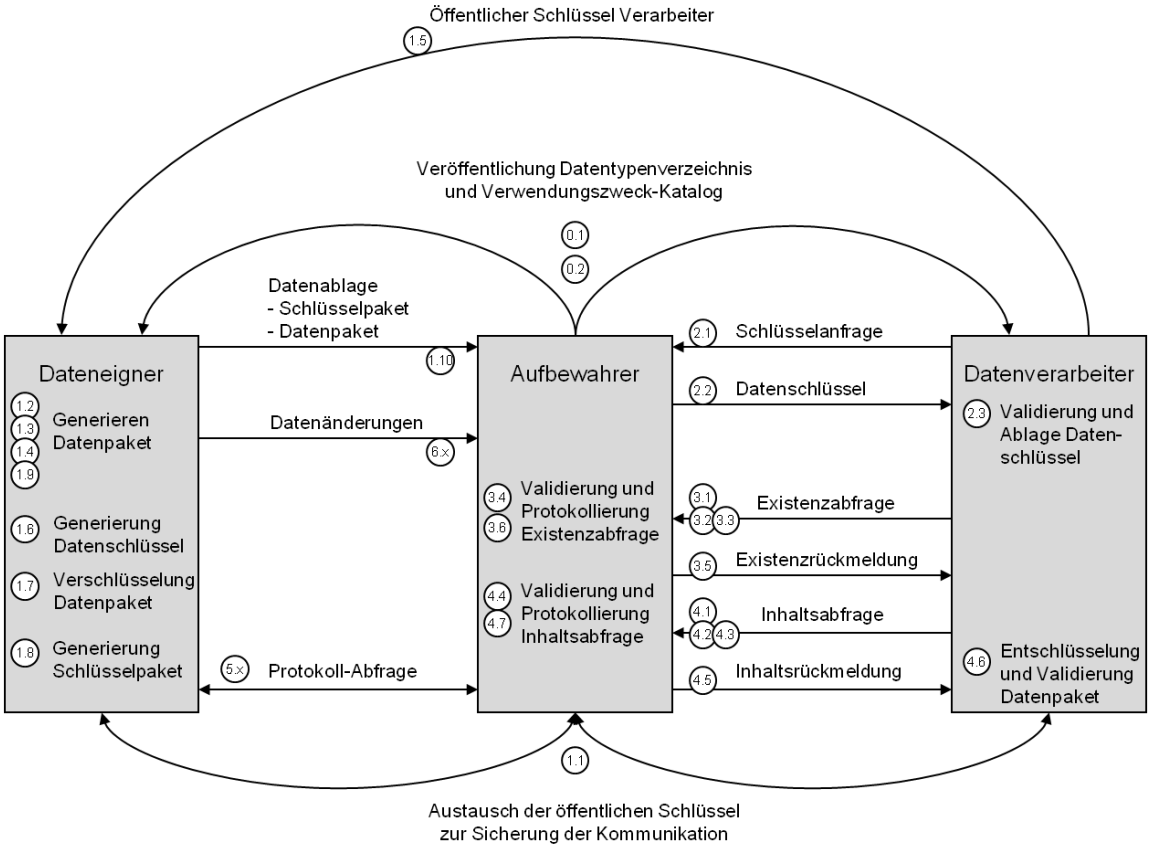


Abbildung 14: Schema für Szenario 1

5.2.2.1 Hinterlegung der Daten

1.1 Dateneigner A_i und Aufbewahrungsstelle T tauschen auf vertrauenswürdigem Weg ihre öffentlichen Schlüssel $K_{pub\ i}$ und $K_{pub\ T}$ aus. Ebenso tauschen T und B_j ihre Schlüssel aus. Dies ist kein eigentlicher Bestandteil des Protokolls, sondern dient der sicheren Kommunikation zwischen den Parteien. Im Folgenden wird davon

ausgegangen, dass alle beteiligten Parteien kryptographisch gesicherte Kommunikationskanäle verwenden[189].

1.2 A_i identifiziert die für ihn relevanten Datensätze sowie die korrespondierenden Datentypen aus der veröffentlichten Liste und ordnet diesen die für ihn gültigen Werte zu. Er erhält die Liste $\{M_{i1}, M_{i2} \ldots M_{ip}\}$, wobei $M_{il} = M(A_i, Y_l)$. Dabei wird vorausgesetzt, dass ein Datensatz eindeutig diese Werte repräsentiert. Für die Kombination {MaxMüller, Religion} gibt es also entweder keinen oder einen Eintrag, keinesfalls aber mehr als einen Eintrag.

Beispiel, dargestellt für Dateneigner 1 (A_1):

Eigner	Datentyp	Daten
A_i	**Y_l**	**M_{il}**
A_1	Y_1	M_{11}
A_1	Y_2	M_{12}
A_1	Y_3	M_{13}
A_1	Y_4	
A_1	Y_5	M_{15}
A_1	Y_6	
…	…	…
A_1	Y_p	M_{1p}

Daten der Typen Y_4 und Y_6 werden in diesem Beispiel nicht bereitgestellt und daher in den folgenden Vorbereitungsarbeiten nicht mehr betrachtet.

1.3 A_i identifiziert die Datenverarbeiter $B_j \in \{B_1, B_2 \ldots B_m\}$, denen jeweils potentiell Zugriff auf mindestens einen der im vorhergehenden Schritt definierten Datensätze gewährt werden soll.

1.4 Den Kombinationen aus Datensatz M_{il} und Verarbeiter B_j ordnet er jeweils mindestens einen erlaubten Verarbeitungszweck J_k zu.

[189] Beispielsweise kommunizieren die Parteien über Transport Layer Security (TLS) bzw. Secure Socket Layer (SSL), das eine Implementierung von Public-Key-Kommunikation darstellt, vgl. etwa [Davi11].

Im Beispiel:

Eigner	Datentyp	Verarbeiter	Zweck
$\mathbf{A_i}$	$\mathbf{Y_l}$	$\mathbf{B_j}$	$\mathbf{J_k}$
A_1	Y_1	B_1	J_5
A_1	Y_2	B_1	J_2
A_1	Y_1	B_2	J_5
A_1	Y_2	B_2	J_2
A_1	Y_2	B_2	J_3
A_1	Y_3	B_3	J_8
A_1	Y_5	B_2	J_3
…	…	…	…
A_1	Y_p	B_m	J_o

Das Ergebnis der bisherigen Schritte lässt sich in einer gemeinsamen Tabelle erfassen. Gemäß dem oben verwendeten Beispiel:

Eigner	Datentyp	Verarbeiter	Zweck	Daten
$\mathbf{A_i}$	$\mathbf{Y_l}$	$\mathbf{B_j}$	$\mathbf{J_k}$	$\mathbf{M_{il}}$
A_1	Y_1	B_1	J_5	M_{11}
A_1	Y_2	B_1	J_2	M_{12}
A_1	Y_1	B_2	J_5	M_{11}
A_1	Y_2	B_2	J_2	M_{12}
A_1	Y_2	B_2	J_3	M_{12}
A_1	Y_3	B_3	J_8	M_{13}
A_1	Y_5	B_2	J_3	M_{15}
…	…	…	…	…
A_1	Y_p	B_m	J_o	M_{1p}

Dabei gilt es zu beachten, dass nur die Kennung des Betroffenen A_i und der Datentyp Y_l den Schlüssel für das eigentliche Datum M_{il} bilden. Der Inhalt hängt nicht vom künftigen Verarbeiter oder dem Zweck der Verarbeitung ab. Es ist nicht wünschenswert, dass ein Dateneigner sich gegenüber verschiedenen Verarbeitern mit unterschiedlichen oder gar

widersprüchlichen Aussagen bzgl. eines Datentyps präsentiert[190]. Wo es notwendig ist, zwischen den an verschiedene Verarbeiter zu übermittelnden Daten eines Typs, etwa hinsichtlich der geforderten Detailtiefe, zu unterscheiden, sind verschiedene Datentypen Y_l zu bilden (z.B.: Konfession / Religion oder: Blutgruppe allgemein / Blutgruppe mit Rhesus-Faktor).

1.5 A_i besorgt sich von entsprechenden Schlüsselservern die öffentlichen Schlüssel aller B_j der Tabelle, d.h. aller Verarbeiter, für die er Daten bereitstellen möchte[191].

1.6 Für jeden dieser Datenverarbeiter erstellt er zudem einen symmetrischen Schlüssel zur gemeinsamen verschlüsselten Kommunikation: $K(A_i, B_j)$, kurz: K_{ij}

Da K_{ij} zur Verschlüsselung der personenbezogenen Daten verwendet wird, sei er im Weiteren als **Datenschlüssel** bezeichnet.

Für jede Kombination von Betroffenem und Verarbeiter stehen somit zwei Schlüssel zur Verfügung, ein symmetrischer Datenschlüssel und der allgemeine öffentliche Schlüssel des Verarbeiters.

Im Beispiel:

Eigner	Verarbeiter	öff. Schlüssel Verarbeiter	Daten-schlüssel
A_i	**B_j**	**$K_{pub\ j}$**	**K_{ij}**
A_1	B_1	$K_{pub\ 1}$	K_{11}
A_1	B_2	$K_{pub\ 2}$	K_{12}
A_1	B_3	$K_{pub\ 3}$	K_{13}
…	…	…	…
A_1	B_m	$K_{pub\ m}$	K_{1m}

1.7 Für jede Zeile der aus Schritt 1.4 resultierenden Tabelle erstellt A_i nun zwei Chiffrate. Als erstes signiert er die Daten mit seinem privaten Schlüssel und chiffriert das Resultat dann mit dem selbst erstellten Datenschlüssel:

$$E_{ij}(S_i(M_{il}))$$

[190] Technisch ist dies dennoch möglich. Der Dateneigner könnte die Schritte des Protokolls auf seiner Seite so modifizieren, dass er die Inhalte zu einem Datentyp für verschiedene Verarbeiter B_j mit unterschiedlichen Werten verschlüsselt. Es gibt Argumente für und gegen den Anspruch, die Eigenverantwortung des Dateneigners über seine Daten technisch zusätzlich zu beschränken. Für die Zuverlässigkeit seiner Angaben gegenüber einem bestimmten Verarbeiter spielt das jedoch keine Rolle. Ein Anreiz für „faires", also einheitliches Verhalten gegenüber allen Verarbeitern kann zumindest die einfachere Übersicht und Wartbarkeit bzgl. der hinterlegten Datensätze sein.

[191] Dies geschieht hier nicht zur nativen Sicherung der Kommunikation wie in Schritt 1, sondern um im Rahmen des Protokolls den kryptographischen Container der hinterlegten Daten erstellen zu können.

Die Signatur der Daten vor der Verschlüsselung dient hier vor allem dem Schutz des Dateneigners vor dem Verarbeiter. Würde das Protokoll ohne Signatur angewendet, könnten Verarbeiter selbst Datenpakete erzeugen und verschlüsseln, sowie Policies zuordnen. Da der von ihnen verwendete Datenschlüssel K_{ij} außer ihnen nur dem Dateneigner bekannt ist, könnte ein Verarbeiter später behaupten, dass die Daten von A_i stammen. Anders herum könnte der Verarbeiter ohne die Signatur keinen Nachweis gegenüber Dritten führen, dass ein Datensatz tatsächlich vom echten Dateneigner für ihn eingestellt wurde.

Als zweites erstellt der Dateneigner die korrespondierenden Policies durch Chiffrieren der Kombination aus Datentyp und Verwendungszweck mit dem Datenschlüssel, den er für den jeweiligen Verarbeiter vorgesehen hat:

$$P_{ijlk} = E_{ij}(Y_l , J_k)$$

Jede Policy kennzeichnet exakt eine Festlegung zwischen einem bestimmten Dateneigner und genau einem Verarbeiter darüber, dass ein Datentyp zu einem bestimmten Zweck abgerufen werden darf. Ihre Eindeutigkeit beruht auf der Chiffrierung mit dem Schlüssel K_{ij}, der nur für je ein Paar aus Dateneigner und -verarbeiter gültig ist, sowie der damit verschlüsselten Kombination aus Datentyp und Verarbeitungszweck.

1.8 Dann signiert und verschlüsselt A_i den symmetrischen Schlüssel K_{ij} wobei E_j die Verschlüsselung mit dem öffentlichen Schlüssel des Verarbeiters B_j darstellt:

$$E_j(S_i(K_{ij}))$$

Die Signatur[192] gibt dem künftigen Verarbeiter die Sicherheit, den Datenschlüssel und damit die mit ihm chiffrierten Datenpakete tatsächlich vom Dateneigner zu erhalten. Dadurch wird ein möglicher Man-in-the-Middle-Angriff verhindert.

[192] Würde zuerst chiffriert und danach signiert, könnte ein Dritter die Signatur unbemerkt auswechseln, daher wird zunächst signiert und dann verschlüsselt. Zu weiteren Aspekten diesbezüglich siehe [Schn96] S.49f.

Es resultiert im Beispiel die folgende Gesamttabelle[193]:

Eigner	Datentyp	Verarbeiter	Zweck	Daten	Daten verschlüsselt	Policy verschlüsselt	Datenschlüssel verschlüsselt
A_i	**Y_l**	**B_j**	**J_k**	**M_{il}**	**$E_{ij}(S_i(M_{il}))$**	**$E_{ij}(Y_l , J_k)$**	**$E_j(S_i(K_{ij}))$**
A_1	Y_1	B_1	J_5	M_{11}	$E_{11}(S_1(M_{11}))$	$E_{11}(Y_1 , J_5)$	$E_1(S_1(K_{11}))$
A_1	Y_2	B_1	J_2	M_{12}	$E_{11}(S_1(M_{12}))$	$E_{11}(Y_2 , J_2)$	$E_1(S_1(K_{11}))$
A_1	Y_1	B_2	J_5	M_{11}	$E_{12}(S_1(M_{11}))$	$E_{12}(Y_1 , J_5)$	$E_2(S_1(K_{12}))$
A_1	Y_2	B_2	J_2	M_{12}	$E_{12}(S_1(M_{12}))$	$E_{12}(Y_2 , J_2)$	$E_2(S_1(K_{12}))$
A_1	Y_2	B_2	J_3	M_{12}	$E_{12}(S_1(M_{12}))$	$E_{12}(Y_2 , J_3)$	$E_2(S_1(K_{12}))$
A_1	Y_3	B_3	J_8	M_{13}	$E_{13}(S_1(M_{13}))$	$E_{13}(Y_3 , J_8)$	$E_3(S_1(K_{13}))$
A_1	Y_5	B_2	J_3	M_{15}	$E_{12}(S_1(M_{15}))$	$E_{12}(Y_5 , J_3)$	$E_2(S_1(K_{12}))$
…	…	…	…	…	…	…	…
A_1	Y_p	B_m	J_o	M_{1p}	$E_{1m}(S_1(M_{1p}))$	$E_{1m}(Y_p , J_o)$	$E_m(S_1(K_{1m}))$

Die vollständige Tabelle wird von A_i erstellt und ist nur ihm bekannt.

1.9 Für jede Zeile der soeben entstandenen Tabelle erzeugt A_i eine Zufallszahl r_{ix}, die er mit der jeweiligen Zeile verknüpft.

Die Zufallszahl spielt keine Rolle für die initiale Ablage von Daten. Sie wird vielmehr später benötigt, wenn der Dateneigner einzelne Datensätze löschen oder aktualisieren will. Dann kann er durch Kenntnis dieser Zufallszahl nachweisen, dass ein Datensatz von ihm erstellt wurde, ohne dass er seine Identität preisgibt.

1.10 Er übermittelt nun voneinander getrennt zwei Strukturen an den Aufbewahrer T, wobei die Übermittlung mit dem öffentlichen Schlüssel von T chiffriert ist:

Schlüsselpaket (Szenario 1)

Verarbeiter	Datenschlüssel verschlüsselt
B_j	**$E_j(S_i(K_{ij}))$**
B_1	$E_1(S_1(K_{11}))$
B_2	$E_2(S_1(K_{12}))$
B_3	$E_3(S_1(K_{13}))$
…	…
B_m	$E_m(S_1(K_{1m}))$

[193] Zur besseren Anschaulichkeit der Darstellung ist die Tabelle hier nicht normalisiert.

Datenpaket (Szenario 1)

Daten verschlüsselt (aus Schritt 1.7)	Policy verschlüsselt (aus Schritt 1.7)	Zufallszahl (aus Schritt 1.9)
$\mathbf{E_{ij}(S_i(M_{il}))}$	$\mathbf{E_{ij}(Y_l , J_k)) = P_{ijlk}}$	$\mathbf{r_{ix}}$
$E_{11}(S_1(M_{11}))$	$E_{11}(Y_1 , J_5)$	r_{11}
$E_{11}(S_1(M_{12}))$	$E_{11}(Y_2 , J_2)$	r_{12}
$E_{12}(S_1(M_{11}))$	$E_{12}(Y_1 , J_5)$	r_{13}
$E_{12}(S_1(M_{12}))$	$E_{12}(Y_2 , J_2)$	r_{14}
$E_{12}(S_1(M_{12}))$	$E_{12}(Y_2 , J_3)$	r_{15}
$E_{13}(S_1(M_{13}))$	$E_{13}(Y_3 , J_8)$	r_{16}
$E_{12}(S_1(M_{15}))$	$E_{12}(Y_5 , J_3)$	r_{17}
…	…	…
$E_{1m}(S_1(M_{1p}))$	$E_{1m}(Y_p , J_o)$	r_{1z}

Durch die gemeinsame Ablage des verschlüsselten Datums $E_{ij}(S_i(M_{il}))$ und der ebenfalls verschlüsselten Policy $E_{ij}(Y_l, J_k) = P_{ijlk}$ wird das Konzept der „Sticky Policies“ für die Aufbewahrung bei T umgesetzt[194].

Weder die Policy, noch die eigentlichen Daten sind für T lesbar. Auch hat er keine realistische Option, die Schlüssel K_{ij} aufzudecken, wenn man für die Verschlüsselung E einen hinreichend sicheren asymmetrischen Algorithmus unterstellt und B_j seinen privaten Schlüssel zuverlässig geheim hält.

Gelingt es dem Dateneigner A_i seine Schlüssel- und Datenpakete so über ein Anonymisierungsnetzwerk (z.B. AN.ON[195] oder JonDo[196]) zu übermitteln, dass er als Absender nicht identifizierbar ist, wird weder T noch sonst eine Partei außer dem künftigen Empfänger erfahren, wer die Daten hinterlegt hat. Das Datenpaket erhält keine erkennbaren Hinweise auf die Quelle, solange der Beobachter nicht über den Datenschlüssel K_{ij} verfügt, mit dem sowohl die Policy als auch der Datensatz verschlüsselt sind. Das Schlüsselpaket lässt zwar den Empfänger, nicht jedoch den Absender erkennen, da die Signatur des Absenders nur für denjenigen prüfbar ist, der den passenden privaten Schlüssel besitzt, also B_j.

[194] vgl. u.a. [AsPS02], [Möll06] S.100
[195] http://www.anon-online.de (Zugriff: 08.07.2012).
[196] http://www.anonym-surfen.de/jondo.html (Zugriff: 08.07.2012).

5.2.2.2 Schlüsselabfrage

Alle Existenz- und Inhaltsabfragen, die ein Datenverarbeiter B_j an die Aufbewahrungsstelle T richtet, beruhen auf einem durch den Dateneigner festgelegten symmetrischen Schlüssel (der Datenschlüssel). Dieser dient als wesentliche Grundlage der Geheimhaltung von Inhalten vor dem Aufbewahrer selbst und anderen möglichen Lauschern.

Der symmetrische Schlüssel K_{ij} wurde von A_i erstellt, als er die Datensätze zur Aufbewahrung zusammengestellt hat. K_{ij} darf nur dem spezifischen Dateneigner A_i und dem von ihm ausgewählten potentiellen Verarbeiter B_j bekannt sein. Dazu ist anzumerken, dass B_j die Kenntnis des Schlüssels K_{ij} noch keine weiteren Daten über A_i verrät. B_j kann die Schlüsselabfrage zu einem beliebigen Zeitpunkt zwischen der Ablage durch A_i und der ersten Existenz- oder Inhaltsabfrage durch ihn selbst durchführen. Da der Schlüssel für die Kombination (A_i,B_j) eindeutig und zugleich unabhängig von Datentyp und Verwendungszweck ist, muss die Schlüsselabfrage nur einmal pro Dateneigner-Verarbeiter-Paar durchgeführt werden. In der Praxis wird B_j immer alle für ihn hinterlegten Schlüsselpakete beim Aufbewahrer abfragen, wenn er den Datenschlüssel zu einem Eigner benötigt, den er noch nicht in seiner Schlüsseldatenbank findet. Er fragt alle Schlüssel ab, weil der Aufbewahrer keine Möglichkeit hat, die Schlüsselpakete zu identifizieren, die von einem bestimmten Eigner abgelegt wurden. Erst durch das Entschlüsseln der Schlüsselpakete mit seinem privaten Schlüssel und dem anschließenden Prüfen der Signaturen werden die Schlüsselpakete für B_j den verschiedenen A_i, die Daten für ihn hinterlegt haben, zuordenbar.

2.1 B_j richtet an den Aufbewahrer T eine elektronische Anfrage mit der Kennzeichnung, dass es sich um eine Schlüsselabfrage handelt, sowie dem Parameter B_j, also seiner eigenen Kennung. T akzeptiert die Anfrage nur, wenn sie durch B_j signiert ist:

$$S_j(\text{„Schlüsselabfrage“}, B_j)$$

2.2 Der Aufbewahrer antwortet entweder mit einem Nullwert im Fall der Nicht-Existenz des angeforderten Schlüssels in seiner Datenbank bzw. im Erfolgsfall mit dem entsprechenden Datensatz $E_j(S_i(K_{ij}))$. Für T ist dieser Datensatz nicht lesbar, da A_i ihn vor der Ablage mit dem öffentlichen Schlüssel des B_j verschlüsselt hat.

2.3 B_j jedoch kann auf $E_j(S_i(K_{ij}))$ seinen privaten Schlüssel anwenden, um den gemeinsamen Datenschlüssel zwischen ihm und A_i im Klartext zu erhalten, sowie die Signatur des A_i zu validieren:

$$K_{ij} = V_i(D_j(E_j(S_i(K_{ij}))))$$

B_j legt den ermittelten Datenschlüssel in einem nur ihm zugänglichen Schlüsselspeicher ab, von wo er ihn für die künftige Nutzung bei allen Anfragen, die sich auf A_i beziehen, abruft.

2.4 T versieht die Schlüsselabfrage mit einem Zeitstempel und speichert sie zusammen mit der Antwort in seiner Protokolldatei.

5.2.2.3 Existenzanfrage

Für einen Verarbeiter ist es mitunter wichtig zu wissen, ob für ihn ein bestimmter Datensatz hinterlegt ist. Den Inhalt will er zu diesem Zeitpunkt noch nicht lesen, weil der konkrete Zweck für die Verarbeitung noch nicht vorliegt. Um die Existenz eines Datensatzes bei der Aufbewahrungsstelle in Erfahrung zu bringen, sieht das System eine „Existenzanfrage“ vor. Hierbei wird durch den Aufbewahrer T auf Anforderung durch B_j geprüft, ob ein Datensatz vorliegt, der

- durch A_i für B_j verschlüsselt wurde, sowie
- einer bestimmten Policy, also Kombination von Datentyp Y_l und erlaubtem Verarbeitungszweck J_k entspricht.

Der Aufbewahrer erlangt durch diese Abfrage keine direkten zusätzlichen Informationen. Sowohl die Policy als auch der Inhalt der Datensätze bleiben ihm verborgen. Er könnte jedoch versuchen, aus der zeitlichen Abfolge der stattfindenden Kommunikation und den gewählten Übertragungswegen Schlüsse zu ziehen. Beides kann durch Anwendung zusätzlicher Sicherungsmaßnahmen wie Anonymisierungsdiensten jedoch stark beschränkt werden.

3.1 B_j bestimmt zunächst die Kombination von Datentyp Y_l und dem angestrebtem Verwendungszweck J_k.

3.2 Diese verschlüsselt er mit dem zuvor erhaltenen symmetrischen Schlüssel: $E_{ij}(Y_l, J_k)$. Dadurch erstellt er einen Vergleichswert $P_{ijlk}´$ für die von A_i hinterlegte Policy P_{ijlk}.

3.3 B_j übermittelt dieses Chiffrat zusammen mit der Kennzeichnung, dass es sich um eine Existenzanfrage handelt, und von ihm signiert, an T.

$$S_j(\text{„Existenzanfrage“}, P_{ijlk}´)$$

3.4 T überprüft die Signatur und vergleicht anschließend die für ihn nichtssagende Ziffernfolge $P_{ijlk}´ = E_{ij}(Y_l, J_k)$ mit allen Einträgen P_{ijlk} seiner Datenbank.

Da T den Schlüssel K_{ij} nicht kennt, mit dem die Anfrage verschlüsselt wurde, kann er keine Rückschlüsse auf deren Inhalt ziehen. Er weiß nur, dass die Anfrage von B_j kommt. Auch kann er aus der Anfrage nicht ermitteln, von wem die Policy und der dazu hinterlegte Datensatz stammen.

3.5 T übermittelt das Ergebnis des Vergleichs an B_j. Findet er einen Eintrag, für den gilt $P_{ijlk} = P_{ijlk}´$, übermittelt er „Ja“, anderenfalls „Nein“. Der Verarbeiter weiß nun,

ob A_i Daten für ihn und für die von ihm gegebenenfalls künftig benötigte Kombination von Zweck und Datentyp hinterlegt hat.

3.6 T versieht die Existenzanfrage mit einem Zeitstempel und speichert sie zusammen mit der Antwort in seiner Protokolldatei.

5.2.2.4 Inhaltsabfrage

Unabhängig davon, ob zuvor eine Existenzanfrage für eine bestimmte Kombination (A_i, B_j, Y_l, J_k) stattgefunden hat, tritt irgendwann der Zeitpunkt ein, zu dem B_j einen bestimmten Datensatz von A_i inhaltlich zur Kenntnis nehmen muss.

Dazu muss er den Datentyp qualifizieren, den er abrufen möchte, und die Anfrage mit einem der vereinbarten Verarbeitungszwecke verknüpfen. Es sei daran erinnert, dass jeder A_i bei Ablage seiner Daten alle Sätze für die jeweiligen Verarbeiter B_j mit einem oder mehreren zugelassenen Zwecken pro Datentyp versehen hat. Nur wenn B_j für die angeforderten Daten einen Verarbeitungszweck angibt, der dem Vergleich mit Datentyp und Zweck, wie bei T abgelegt, standhält, wird er den entsprechenden Datensatz übermittelt bekommen.

4.1 B_j bestimmt zunächst die Kombination von Datentyp Y_l und von ihm angestrebtem Verwendungszweck J_k.

4.2 Diese verschlüsselt er mit dem zuvor erhaltenen symmetrischen Schlüssel zu $E_{ij}(Y_l, J_k)$ und erhält analog zur Existenzanfrage den Policy-Vergleichswert $P_{ijlk}´$.

4.3 B_j übermittelt $P_{ijlk}´$ zusammen mit der Kennzeichnung, dass es sich um eine Datenabfrage handelt, und von ihm signiert, an T.

$$S_j(\text{„Inhaltsabfrage“}, P_{ijlk}´)$$

4.4 T überprüft die Signatur und anschließend die für ihn nichtssagende Ziffernfolge $P_{ijlk}´ = E_{ij}(Y_l, J_k)$ mit allen Einträgen P_{ijlk} seiner Datenbank[197]. Auch hier gilt wieder: Da T den Schlüssel K_{ij} nicht kennt, mit dem die Anfrage verschlüsselt wurde, kann er keine Rückschlüsse auf deren Inhalt ziehen.

4.5 Hat T einen Datensatz mit $P_{ijlk} = P_{ijlk}´$ gefunden, übermittelt er das korrespondierende Datenpaket $E_{ij}(S_i(M_{il}))$ an B_j. Ihm selbst wird dabei der Inhalt des Pakets nicht bekannt. Existiert kein korrespondierendes Paket zur Policy $P_{ijlk}´$ erhält der Verarbeiter analog zur Existenzanfrage ein „Nein“ zur Antwort.

[197] Durch den Vergleich auf Identität folgt er der harten Forderung des BDSG. „Ausschlaggebendes Kriterium für die Zulässigkeit einer Datenverarbeitung ist nach dieser Konzeption (des BDSG – Anmerkung des Autors) folglich der Zweck, zu dem die Daten erhoben worden sind. Nur wenn die Zwecke identisch sind, darf eine Verarbeitung erfolgen. Insoweit ist das deutsche Datenschutzrecht strenger als das europäische Datenschutzrecht, das in Art. 6 Abs. 1b DSRL 95/46/EG nur die Vereinbarkeit der Zwecke verlangt.“ [FoKr05]

4.6 B_j kann im Erfolgsfall das Paket mit Hilfe des Schlüssels K_{ij} in den Klartext umsetzen und die Signatur von A_i mit dessen öffentlichem Schlüssel prüfen:

$$M_{il} = V_i(D_{ij}(E_{ij}(S_i(M_{il}))))$$

4.7 T erstellt über die Herausgabe des Datensatzes einen Logeintrag, den er zusammen mit der zugehörigen Anfrage durch B_j und einem Zeitstempel speichert. A_i kann später anhand dieser Logs nachvollziehen, wer zu welchem Zeitpunkt mit welchem Zweck welchen Datentyp wann abgerufen hat.

Die Aufbewahrungsstelle protokolliert alle erfolgreichen und fehlgeschlagenen Abfragen, die der Betroffene jederzeit einsehen kann.

5.2.2.5 Logabfrage

Jeder Dateneigner A_i kann auf Anfrage das Protokoll der für seine Datensätze gestellten Existenz- und Inhaltsabfragen vom Aufbewahrer abrufen. Dazu muss er seine Identität nicht preisgeben[198].

5.1 A_i wählt aus den von ihm initial erstellten verschlüsselten Policies $P_{ijlk} = E_{ij}(Y_l, J_k)$ diejenigen aus, für die er die Anfrageprotokolle abrufen möchte.

5.2 Er übermittelt die Liste dieser Policies P_{ijlk}´´mit der Kennzeichnung, dass es sich um eine Logabfrage handelt an T. Um zu vermeiden, dass T aus der Gesamtanfrage erkennt, welche Datensätze zu einem gemeinsamen Dateneigner gehören, übermittelt er die Anfragen mit zufälligem zeitlichen Abstand und auf unterschiedlichen Wegen.

5.3 Für jede Logabfrage mit einer übermittelten Policy P_{ijlk}´´ durchsucht T seine Logtabelle und erstellt eine Liste aller Existenz- und Inhaltsabfragen, für die P_{ijlk}´´ = P_{ijlk}.

5.4 Er stellt A_i die entstandene Liste zur Verfügung. A_i kann nun jede Anfrage, die sich auf seine Daten bezieht, mit dem entsprechenden Zeitstempel überprüfen. Hält er eine Anfrage für nicht gerechtfertigt, kann er entsprechende Schritte gegen den Verarbeiter einleiten. Die Verarbeiter können nicht abstreiten, die Anfragen gestellt zu haben, da diese mit ihrer persönlichen digitalen Signatur versehen sind.

Alternativ zu Anfrage und Antwort könnte T auch eine jeweils aktuelle Liste aller Anfragen auf seiner Website veröffentlichen. Dazu genügt die Angabe, ob es sich um eine Existenz- oder Inhaltsabfrage gehandelt hat, der Zeitstempel und die verschlüsselte Policy P_{ijlk}. Die Kennung des Anfragenden B_j muss nicht aufgelistet werden, da die jeweiligen

[198] Dies setzt voraus, dass er seine Anfrage auch technisch anonymisiert stellt. Im einfachsten Fall nutzt er eine Einmal-E-Mail-Adresse für die Kommunikation mit T. Bei einer Abfrage etwa über ein Webformular, muss er dafür sorgen, dass seine IP-Adresse verschleiert wird.

Dateneigner diese aus dem Vergleich mit den von ihnen erstellten Policy-Chiffraten selbst ermitteln können. Die Datenverarbeiter B_j können anhand der veröffentlichten Liste und ihrer eigenen Aufzeichnungen zugleich überprüfen, ob T ihre Anfragen korrekt protokolliert hat.

Da in diesem Falle die Veröffentlichung ohne die Signatur des Datenverarbeiters erfolgt, hat ein Dateneigner zunächst kein Beweismittel an der Hand, durch das er einem B_j eindeutig nachweisen kann, dass eine Anfrage von ihm stammt. B_j könnte behaupten, A_i oder T hätten die Anfrage eingeschleust. Bei einem solchen Disput kann sich A_i jedoch an T wenden und durch ihn die vollständige Original-Anfrage inklusive der Signatur des B_j offenlegen lassen. Mit dieser lässt sich sodann beweisen, dass die Anfrage tatsächlich von B_j kam, da nur dieser im Stande ist, die Signatur zu erzeugen.

5.2.2.6 Datenänderungen durch den Eigner

Personenbezogene Daten ändern sich im Verlauf der Zeit. Familienstatus, Religionszugehörigkeit, Krankendaten ebenso wie persönliche Interessen oder erwünschte Interaktionspartner. Dementsprechend ist im Protokoll die Möglichkeit vorzusehen, Änderungen, Ergänzungen und Löschungen an den Daten vorzunehmen, die bei der Aufbewahrungsstelle hinterlegt sind.

Verschiedene Aktionen sind aus Sicht des Dateneigners möglich:

6.1 Generelles Ausscheiden des Dateneigners aus dem System: A_i belässt die Schlüsselpakete beim Aufbewahrer. Um den Verarbeitern den Zugriff auf die hinterlegten Daten zu entziehen, lässt A_i alle Sätze seiner hinterlegten Datenpakete löschen, wie im folgenden Schritt beschrieben[199].

6.2 Entfernen von Inhalten: Möchte A_i einen Datensatz bei T löschen, muss er nachweisen, dass dieser Datensatz von ihm selbst eingestellt wurde. Schließlich will man niemandem erlauben, die Datensätze anderer Eigner zu manipulieren. Da das Einstellen der Daten anonym geschah und A_i auch jetzt seine Identität nicht preisgeben will, wird ein Weg benötigt, seine Privatheit zu wahren und gleichzeitig T davon zu überzeugen, dass die eingegangene Löschanforderung legitim ist.

A_i tritt diesen Beweis gegenüber T dadurch an, dass er sein Wissen der Zufallszahl r_{ix} offenbart, die dem zu löschenden Datensatz zugeordnet ist. Zur Erinnerung: A_i hatte bei der initialen Anlage zu jeder Zeile des Datenpakets eine Zufallszahl r_{ix} erzeugt, diese mit den Inhaltsdaten sowie der Policy verknüpft und gemeinsam mit diesen an T übermittelt, wobei das Gesamtpaket mit dem öffentlichen Schlüssel von T chiffriert war. Außer A_i und T kann also niemand die Zahl r_{ix} kennen. Wenn

[199] Wird Wert auf ein Entfernen der Schlüsselpakete gelegt, ist dies analog zu den Datenpaketen durch die initiale Zuweisung einer Zufallszahl und den späteren Nachweis der Kenntnis dieser Zahl realisierbar, ohne dass die Identität des Eigners gegenüber dem Aufbewahrer aufgedeckt werden müsste.

eine Löschanforderung für einen Datensatz gemeinsam mit einer Referenz auf die korrespondierende Zufallszahl bei T eingeht, kann dieser davon ausgehen, dass die Anforderung von demjenigen stammt, der auch das ursprüngliche Datenpaket zusammengestellt hat.

Um Replay-Angriffe zu verhindern, übermittelt A_i jedoch nicht die initial erzeugte Zufallszahl. Er führt stattdessen zunächst eine Einweg-Hash-Funktion über r_{ix} ($h(r_{ix})$) durch und übermittelt das Resultat zusammen mit der Policy als Identifikator des zu löschenden Datensatzes. Er verschlüsselt die Gesamtnachricht mit dem öffentlichen Schlüssel der Aufbewahrungsstelle. Die Nachricht lautet also:

$$E_T(\text{„Löschen“}, P_{ijlk}, h(r_{ix}))$$

T ermittelt nun seinerseits $r_{ix}' = h(r_{ix}))$ für das r_{ix}, das durch die initiale Datenablage dem über P_{ijlk} identifizierten Datensatz zugeordnet ist. Wenn der von ihm berechnete Wert mit dem übermittelten Wert übereinstimmt, kann er davon ausgehen, dass die Löschanforderung vom legitimen Dateneigner stammt und löscht den entsprechenden Datensatz aus seiner Datenbank.

6.3 Hinzufügen eines Empfängers und entsprechender Daten: Das Vorgehen ist analog zur initialen Datenablage. A_i sendet getrennt voneinander ein zusätzliches Schlüsselpaket und korrespondierende Datenpakete.

6.4 Hinzufügen von Inhalten[200]: A_i sendet anonym an T zusätzliche Datenpakete.

6.5 Ausschließen eines Empfängers: Soll ein einzelner Verarbeiter für alle Daten eines Eigners ausgeschlossen werden, lässt er alle für B_j hinterlegten Einträge im Datenpaket löschen, wie in „Entfernen von Inhalten“ beschrieben.

6.6 Änderung bestehenden Inhalts: Es besteht prinzipiell dasselbe Problem wie bei der Löschung von Datensätzen. A_i muss gegenüber T nachweisen, dass er der Dateneigner eines Satzes ist, ohne dass er seine Identität preisgibt. Analog zur Löschung dient die initial mit den Datensätzen assoziierte Zufallszahl r_{ix} als Beweis, dass ein Datensatz von demjenigen stammt, der nun eine Aktualisierung veranlassen möchte. Wieder übermittelt der Dateneigner die Policy als Identifikator des Datensatzes und den Hashwert über die korrespondierende Zahl r_{ix}. Im Gegensatz zur Löschanforderung nimmt er nun auch zusätzlich den veränderten Datensatz $E_{ij}(S_i(M_{il}))'$ auf, der anstelle des bisherigen $E_{ij}(S_i(M_{il}))$ bei T zur Policy P_{ijlk} hinterlegt werden soll. Die Nachricht lautet:

$$E_T(\text{„Änderung“}, P_{ijlk}, E_{ij}(S_i(M_{il}))', h(r_{ix}))$$

[200] Unter dem Hinzufügen von Inhalten wird das Erstellen einer neuen Kombination von Policy und Daten für einen Verarbeiter verstanden. Im Gegensatz dazu steht die Erweiterung der Daten zu einer bereits hinterlegten Policy (z.B. Ergänzen der Krankengeschichte um zusätzliche Fakten). Diese werden unter „Änderung bestehenden Inhalts“ behandelt.

Anschließend ersetzen sowohl A_i als auch T in ihren Datentabellen den in der geänderten Zeile gespeicherten Wert r_{ix} durch $h(r_{ix})$. Für alle künftigen Änderungsanforderungen gilt dann $r_{ix}' = h(r_{ix})$. T akzeptiert nur Lösch- oder Änderungsanforderungen für einen Datensatz, wenn die mitgelieferte Zahl r_{ix} dem Ergebnis der nächsten Durchführung des Hash-Algorithmus entspricht. Auf diese Weise ist das System vor Replay-Angriffen durch Dritte geschützt.

6.7 Änderung der Zuweisung von Datentyp oder Verarbeitungszweck: Diese Änderungen sind wie die Löschung eines Datensatzes und die Lieferung eines neuen Pakets zu behandeln. Entsprechend sind die Aktionen analog auszuführen.

Entfernt A_i Datensätze aus der Ablage bei T, sei es durch generelles Ausscheiden aus dem System oder durch gezielte Löschanforderungen für einzelne Datensätze, kann es sein, dass Verarbeiter, die bereits in der Vergangenheit eine erfolgreich beantwortete Existenzanfrage gestellt haben, nun im falschen Glauben sind, der damals angefragte Datensatz stünde für künftige Verarbeitungen noch zur Verfügung. Sie müssen informiert werden, wenn dies durch die Löschungen nicht mehr zutrifft.

6.8 Also prüft T alle historischen Existenzanfragen aus seinem Protokoll erneut, wenn eine Löschanforderung durch den Dateneigner initiiert wurde. Dazu vergleicht er die von A_i als Identifikator übermittelten P_{ijlk} mit den im Protokoll als abgefragt gekennzeichneten Policies und informiert bei Übereinstimmung den jeweiligen Verarbeiter B_j.[201]

[201] Die Änderung bestehenden Inhalts wird nicht einfach als Löschung eines Datensatzes und Hinzufügen eines Neuen realisiert. Denn in diesem Fall würde sie eine Löschbenachrichtigung erzeugen, die für reine Inhaltsänderungen nicht gewünscht ist.

5.2.3 Protokolle für Szenario 2

Ergänzend zu den Definitionen aus Szenario 1 sei auch:

Empfängergruppe $j \in \{1, 2, \ldots, m\}$ mit m = Anzahl der unterschiedlichen Gruppen potentieller Empfänger, die Daten der Individuen abrufen	B_{Gj}
Zertifizierungsinstanz / Zertifizierer	Z
Symmetrischer Schlüssel (zwischen A_i und B_{Gj})	$K(A_i, B_{Gj}) = K_{iGj}$
Verschlüsselung von M mit dem symmetrischen Schlüssel K_{iGj}	$E_{iGj}(M) = C$
Entschlüsseln von C mit dem symmetrischen Schlüssel K_{iGj}	$D_{iGj}(C) = M$
Policy für Empfängergruppen, d.h. eine Kombination von Datensatztyp Y_l und Verarbeitungszweck J_k, verschlüsselt mit K_{iGj}	$E_{iGj}(Y_l, J_k) = P_{iGjlk}$
One-Time-Pad für den Schlüsselaustausch • zwischen A_i und Z für Empfängergruppe B_{Gj} • zwischen Aufbewahrer T und dem Verarbeiter B_j	$OTP(A_i, B_{Gj}) = OTP_{iGj}$ $OTP(T, B_j) = OTP_{Tj}$
Anwendung des One-Time-Pad OTP_{xy} auf die Bitfolge z durch bitweises Exklusiv-ODER	$OTP_{xy}(z)$
Anwendung eines Krypto-Protokolls, das einem Mitglied der Empfängergruppe B_{Gj} den Schlüssel x geheim und asynchron übermittelt.	$E_{Gj}(x)$

Zwar weiß der Betroffene, dass er bestimmten Gruppen von Verarbeitern Zugriff auf Daten gewähren möchte, aber er kennt vorab nicht die Mitglieder der Gruppe, etwa weil die Gruppe auch nach Ablage der Daten noch um zusätzliche Mitglieder wachsen wird.

Anders als in Szenario 1 sind dem Betroffenen somit nicht die öffentlichen Schlüssel der künftigen Verarbeiter bekannt. Ersatzweise wird als zusätzliche Partei die Zertifizierungsstelle Z in das Protokoll aufgenommen. Sie hat die Aufgabe, Verarbeiter in ihrer Zugehörigkeit zu bestimmten Gruppen zu zertifizieren und die Gruppenmitgliedschaften für die anderen Parteien zu verwalten. Sie dient so auch als Mittler in der Datenweitergabe.

Abbildung 15 zeigt die wesentlichen Unterschiede zu Szenario 1. Die Nummerierung entspricht den im Folgenden geschilderten Schritten. Die Abbildung stellt die Schritte bis zur erfolgreichen Schlüsselabfrage (3.4) dar. Für die weiteren Schritte – Existenz- und

Inhaltsabfragen, Protokollabfrage und Datenänderungen durch den Dateneigner – ist der Ablauf analog zu Szenario 1.

Die Schritte zur Schlüsselabfrage sind in der Abbildung durch dicke gestrichene Pfeile hervorgehoben. Sie verdeutlichen das zugrunde liegende Prinzip der Verantwortungstrennung: Dateneigner und Aufbewahrer müssen den Zertifizierer mit den entsprechenden Daten versorgen (das gruppenspezifische One-Time-Pad vom Dateneigner und später den mit beiden OTP chiffrierten Datenschlüssel vom Aufbewahrer), damit der Zertifizierer den Datenverarbeiter mit Paketen versorgen kann. Ebenso ist der Verarbeiter auf Daten von Aufbewahrer und Zertifizierer angewiesen (den mit dem empfängerspezifischen OTP chiffrierten Datenschlüssel vom Zertifizierer und das dazu passende OTP vom Aufbewahrer), um schließlich an die nutzbaren Daten zu gelangen.

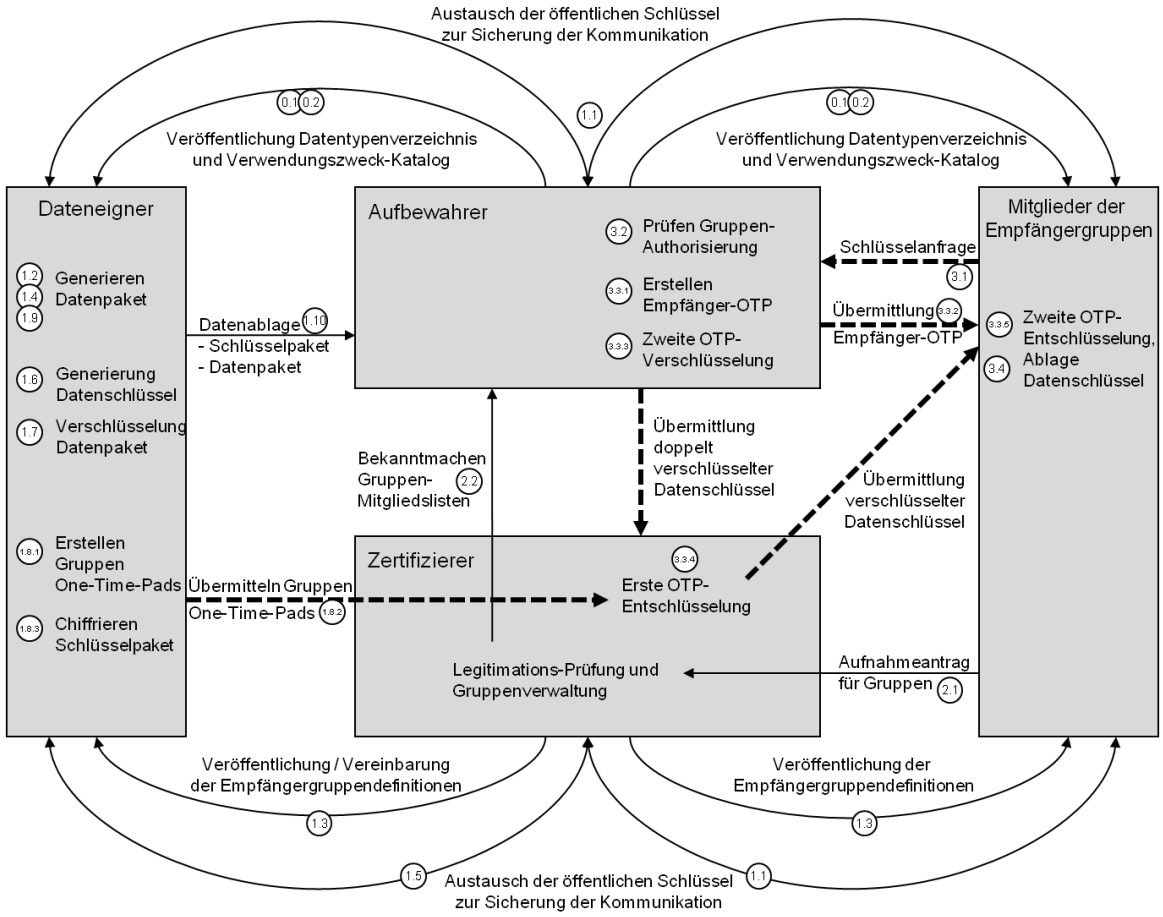


Abbildung 15: Schema für Szenario 2

5.2.3.1 Hinterlegung der Daten

Unverändert bleiben die Kataloge, in denen die möglichen Zwecke der Datenverarbeitung J_k und die auswählbaren Datensatz-Typen Y_l veröffentlicht werden. Ebenfalls identisch zum Protokoll von Szenario 1 werden die grundlegenden Datensätze durch den Eigner erstellt:

1.1 Alle beteiligten Parteien tauschen auf vertrauenswürdigem Weg ihre öffentlichen Schlüssel aus um alle folgende Kommunikation grundlegend zu schützen.

1.2 A_i identifiziert wie in Szenario 1 die für ihn relevanten Datensätze sowie die korrespondierenden Datentypen aus der veröffentlichten Liste und ordnet diesen die für ihn gültigen Werte zu. Er erhält die Liste $\{M_{i1}, M_{i2} \dots M_{ip}\}$, wobei $M_{il} = M(A_i, Y_l)$.

Beispiel, dargestellt für Dateneigner 1 (A_1):

Eigner	Datentyp	Daten
$\mathbf{A_i}$	$\mathbf{Y_l}$	$\mathbf{M_{il}}$
A_1	Y_1	M_{11}
A_1	Y_2	M_{12}
A_1	Y_3	M_{13}
A_1	Y_4	
A_1	Y_5	M_{15}
A_1	Y_6	
…	…	…
A_1	Y_p	M_{1p}

1.3 A_i kennt die künftigen Datenverarbeiter in Szenario 2 nicht, er kann nur Gruppen von Verarbeitern festlegen, denen er Zugriff auf im vorhergehenden Schritt definierte Datensätze gewähren möchte. Anders als in Szenario 1 identifiziert er also nicht die Datenverarbeiter B_j, sondern Verarbeitergruppen $B_{Gj} \in \{B_{G1}, B_{G2} \dots B_{Gm}\}$.

Die Definition der Gruppen vereinbart A_i mit der neuen Zertifizierungsinstanz Z, oder er übernimmt eine Auswahl der von Z bereits allgemein bereit gestellten Gruppendefinitionen[202].

1.4 Den Kombinationen aus Datensatz M_{il} und Verarbeitergruppe B_{Gj} ordnet er jeweils mindestens einen erlaubten Verarbeitungszweck J_k zu.

In einer gemeinsamen Tabelle erfasst:

Eigner	Datentyp	Verarbeiter-gruppe	Zweck	Daten
A_i	**Y_l**	**B_{Gj}**	**J_k**	**M_{il}**
A_1	Y_1	B_{G1}	J_5	M_{11}
A_1	Y_2	B_{G1}	J_2	M_{12}
A_1	Y_1	B_{G2}	J_5	M_{11}
A_1	Y_2	B_{G2}	J_2	M_{12}
A_1	Y_2	B_{G2}	J_3	M_{12}
A_1	Y_3	B_{G3}	J_8	M_{13}
A_1	Y_5	B_{G2}	J_3	M_{15}
…	…	…	…	…
A_1	Y_p	B_{Gm}	J_o	M_{1p}

1.5 In Szenario 1 besorgt sich A_i an dieser Stelle die öffentlichen Schlüssel aller Verarbeiter B_j für die er Daten bereitstellen möchte. In Szenario 2 kennt er nur die Verarbeitergruppen, deren Mitglieder zu diesem Zeitpunkt noch nicht feststehen. Entsprechend kann auch kein öffentlicher Schlüssel bereitgestellt werden, mit dessen zugeordneten privaten Schlüssel die Datenverarbeiter später den Klartext übermittelter Datensätze wiederherstellen könnten[203].

Hier kommt nun die Zertifizierungsinstanz Z ins Spiel. Sie repräsentiert die künftigen Datenverarbeiter und ihre Gruppen. Entsprechend beschafft A_i an dieser Stelle auch den öffentlichen Schlüssel des Zertifizierers $K_{pub\ Z}$.

[202] Vereinfachend wird hier angenommen, dass die Zusammensetzung der Empfängergruppen zwar variabel ist, d.h. hinsichtlich der Zusammensetzung ihrer Mitgliedsverzeichnisse alle notwendigen Änderungen und Erweiterungen abbildet, aber stets für alle Dateneigner gleichermaßen gültig ist. Das heißt beispielsweise, dass die Liste der Notfallärzte jederzeit um neue Mitglieder erweitert werden kann, dass aber für Dateneigner A_1 zu jedem Zeitpunkt dieselben Mediziner zu dieser Gruppe zählen wie für alle anderen Dateneigner A_2 bis A_n. Diese Annahme kann natürlich dadurch aufgehoben werden, dass in die Definition der Empfängergruppen ein Identifikator für den Dateneigner aufgenommen wird, der die jeweilige Gruppendefinition akzeptiert. Auf die Grundlagen und die Durchführung des Protokolls hätte dies keine Auswirkungen, es würde sich jedoch der Aufwand zur Datensatz- und Schlüsselverwaltung signifikant erhöhen. Da dies mit einer weniger lesbaren Notation einhergehen würde, wird hier auf diese Komplikation verzichtet.

[203] Bestehende Verfahren zur Gruppenverschlüsselung benötigen anders als das hier vorgestellte System eine zentrale Schlüsselverwaltung, vgl. etwa [Cert11].

1.6 Für jede Empfängergruppe erstellt A_i einen symmetrischen Datenschlüssel: $K(A_i, B_{Gj})$, kurz: K_{iGj}

Für jede Kombination von Betroffenem und Empfängergruppe stehen somit zwei Schlüssel zur Verfügung, ein symmetrischer Datenschlüssel und der öffentliche Schlüssel des Zertifizierers, der für jeden Datensatz identisch ist.

Im Beispiel:

Eigner	Verarbeiter-gruppe	öff. Schlüssel Zertifizierer	Daten-schlüssel
A_i	**B_{Gj}**	**$K_{pub\ Z}$**	**K_{iGj}**
A_1	B_{G1}	$K_{pub\ Z}$	K_{1G1}
A_1	B_{G2}	$K_{pub\ Z}$	K_{1G2}
A_1	B_{G3}	$K_{pub\ Z}$	K_{1G3}
…	…	…	…
A_1	B_{Gm}	$K_{pub\ Z}$	K_{1Gm}

1.7 Für jede Zeile der aus Schritt 1.4 resultierenden Tabelle erstellt A_i nun zwei Chiffrate. Als erstes signiert er die Daten mit seinem privaten Schlüssel und chiffriert das Resultat dann mit dem selbst erstellten Datenschlüssel:

$$E_{iGj}(S_i(M_{il}))$$

Als zweites erstellt der Dateneigner die korrespondierenden Policies durch Chiffrieren der Kombination aus Datentyp und Verwendungszweck mit dem Datenschlüssel, den er für die jeweilige Empfängergruppe vorgesehen hat:

$$P_{iGjlk} = E_{iGj}(Y_l, J_k)$$

In Szenario 2 kennzeichnet eine Policy also exakt eine Festlegung zwischen einem bestimmten Dateneigner und dem Zertifizierer, dass eine bestimmte Empfängergruppe einen Datentyp zu einem bestimmten Zweck abrufen darf. Ihre Eindeutigkeit beruht wiederum auf der Chiffrierung mit dem Schlüssel K_{iGj}, der nur für je ein Paar aus Dateneigner und Empfängergruppe gültig ist, sowie der damit verschlüsselten Kombination aus Datentyp und Verarbeitungszweck.

Besondere Bedeutung kommt in Szenario 2 dem Verschlüsselungsverfahren für den nächsten Schritt zu, also dem $E_{Gj}(x)$, das für die sichere Ablage und den Austausch des Datenschlüssels verwendet wird. Während in Szenario 1 einfach nur ein hinreichend sicherer Algorithmus für $E_j(x)$ gewählt werden muss, besteht in Szenario 2 die zusätzliche

Anforderung, dass der Algorithmus kommutative Schlüsselanwendung erlaubt[204]. Die Diskussion geeigneter Algorithmen findet ausführlich in Kapitel 6.1 statt. Das Ergebnis dieser Untersuchung vorwegnehmend wird für die folgenden Schritte eine Kombination aus asymmetrischer Kryptographie und symmetrischer Stromchiffrierung mit dem One-Time-Pad genutzt.

1.8 A_i signiert und verschlüsselt die symmetrischen Schlüssel K_{iGj}:

$$E_{Gj}(S_i(K_{iGj}))$$

Dabei repräsentiert die Verschlüsselung E_{Gj} mehrere Schritte:

1.8.1 A_i erstellt für jede relevante Empfängergruppe B_{Gj} ein One-Time-Pad OTP_{iGj} („Gruppen-OTP"), das mindestens dieselbe Länge hat wie der signierte Datenschlüssel, den A_i für die Mitglieder der Empfängergruppe hinterlegt[205].

1.8.2 A_i verschlüsselt die erstellten One-Time-Pads mit dem öffentlichen Schlüssel des Zertifizierers Z und sendet diesem die Liste der entstandenen $E_Z(OTP_{iGj})$. Z öffnet die Sendung durch Anwendung seines persönlichen privaten Schlüssels $K_{priv\,Z}$ und legt OTP_{iGj} in seinem Schlüsselverwaltungs-Modul ab[206].

Verarbeitergruppe	One-Time-Pad verschlüsselt
B_{Gj}	**$E_Z(OTP_{iGj})$**
B_{G1}	$E_Z(OTP_{1G1})$
B_{G2}	$E_Z(OTP_{1G2})$
B_{G3}	$E_Z(OTP_{1G3})$
…	…
B_{Gm}	$E_Z(OTP_{1Gm})$

1.8.3 A_i verschlüsselt die von ihm signierten Schlüsselpakete $S_i(K_{iGj})$ durch bitweise XOR-Verknüpfung mit den korrespondierenden One-Time-Pads OTP_{iGj}.

$$OTP_{iGj}(S_i(K_{iGj}))$$

[204] Das Protokoll in Szenario 2 ist darauf angewiesen, dass A_i ein Chiffrat herstellt, das für Z dechiffrierbar ist. Dieses Chiffrat wird aber zunächst von T erneut verschlüsselt, und zwar so, dass B_j es später entschlüsseln kann. Die Entschlüsselung insgesamt muss dann zunächst durch Z stattfinden, und anschließend durch B_j. Das bedeutet, die Schlüsselanwendung muss nicht nur nach dem üblichen last-in-first-out (LIFO), sondern auch first-in-first-out (FIFO) funktionieren. Ein Algorithmus, der grundsätzlich kommutative Schlüsselanwendung erlaubt, wird diese spezifische Anforderung in jedem Fall erfüllen.

[205] Kürzere One-Time-Pads würden bedeuten, dass sie zumindest in Teilen mehrfach verwendet würden, was ihre Sicherheit kompromittiert, vgl. [Rijm10a].

[206] Grundsätzlich ist hier jeder Algorithmus einsetzbar, der einen nicht-interaktiven sicheren Austausch symmetrischer Schlüssel gewährleistet.

Es resultiert im Beispiel die folgende Gesamttabelle:

Eigner	Daten-typ	Verar-beiter-gruppe	Zweck	Daten	Daten ver-schlüsselt	Policy verschlüsselt	One-Time-Pad verschlüsselt	Datenschlüssel verschlüsselt
$\mathbf{A_i}$	$\mathbf{Y_l}$	$\mathbf{B_{Gj}}$	$\mathbf{J_k}$	$\mathbf{M_{il}}$	$\mathbf{E_{iGj}(S_i(M_{il}))}$	$\mathbf{E_{iGj}(Y_l , J_k)}$	$\mathbf{E_Z(OTP_{iGj})}$	$\mathbf{OTP_{iGj}(S_i(K_{iGj}))}$
A_1	Y_1	B_{G1}	J_5	M_{11}	$E_{1G1}(S_1(M_{11}))$	$E_{1G1}(Y_1 , J_5)$	$E_Z(OTP_{1G1})$	$OTP_{1G1}(S_1(K_{1G1}))$
A_1	Y_2	B_{G1}	J_2	M_{12}	$E_{1G1}(S_1(M_{12}))$	$E_{1G1}(Y_2 , J_2)$	$E_Z(OTP_{1G1})$	$OTP_{1G1}(S_1(K_{1G1}))$
A_1	Y_1	B_{G2}	J_5	M_{11}	$E_{1G2}(S_1(M_{11}))$	$E_{1G2}(Y_1 , J_5)$	$E_Z(OTP_{1G2})$	$OTP_{1G2}(S_1(K_{1G2}))$
A_1	Y_2	B_{G2}	J_2	M_{12}	$E_{1G2}(S_1(M_{12}))$	$E_{1G2}(Y_2 , J_2)$	$E_Z(OTP_{1G2})$	$OTP_{1G2}(S_1(K_{1G2}))$
A_1	Y_2	B_{G2}	J_3	M_{12}	$E_{1G2}(S_1(M_{12}))$	$E_{1G2}(Y_2 , J_3)$	$E_Z(OTP_{1G2})$	$OTP_{1G2}(S_1(K_{1G2}))$
A_1	Y_3	B_{G3}	J_8	M_{13}	$E_{1G3}(S_1(M_{13}))$	$E_{1G3}(Y_3 , J_8)$	$E_Z(OTP_{1G3})$	$OTP_{1G3}(S_1(K_{1G3}))$
A_1	Y_5	B_{G2}	J_3	M_{15}	$E_{1G2}(S_1(M_{15}))$	$E_{1G2}(Y_5 , J_3)$	$E_Z(OTP_{1G2})$	$OTP_{1G2}(S_1(K_{1G2}))$
…	…	…	…	…	…	…	…	…
A_1	Y_p	B_{Gm}	J_o	M_{1p}	$E_{1Gm}(S_1(M_{1p}))$	$E_{1Gm}(Y_p , J_o)$	$E_Z(OTP_{1Gm})$	$OTP_{1Gm}(S_1(K_{1Gm}))$

Die vollständige Tabelle ist weiterhin nur jeweils demjenigen Eigner A_i bekannt, dessen persönliche Datensätze die einzelnen Zeilen tragen.

1.9 Für jede Zeile der soeben entstandenen Tabelle erzeugt A_i eine Zufallszahl r_{ix}, die er mit der jeweiligen Zeile verknüpft. Auf diese Weise werden die einzelnen Datensätze, die beim Aufbewahrer hinterlegt sind, anonym durch den Eigner adressierbar, wenn spätere Änderungen durchgeführt werden sollen.

1.10 A_i übermittelt nun voneinander getrennt zwei Strukturen an den Aufbewahrer T, wobei die Übermittlung mit dem öffentlichen Schlüssel von T chiffriert ist:

Schlüsselpaket (Szenario 2)

Verarbeiter-gruppe	Datenschlüssel verschlüsselt
$\mathbf{B_{Gj}}$	$\mathbf{OTP_{iGj}(S_i(K_{iGj}))}$
B_{G1}	$OTP_{1G1}(S_1(K_{1G1}))$
B_{G2}	$OTP_{1G2}(S_1(K_{1G2}))$
B_{G3}	$OTP_{1G3}(S_1(K_{1G3}))$
…	…
B_{Gm}	$OTP_{1Gm}(S_1(K_{1Gm}))$

Datenpaket (Szenario 2)

Daten verschlüsselt	Policy verschlüsselt	Zufallszahl
$\mathbf{E_{iGj}(S_i(M_{il}))}$	$\mathbf{E_{iGj}(Y_l , J_k)}$	$\mathbf{r_{ix}}$
$E_{1G1}(S_1(M_{11}))$	$E_{1G1}(Y_1 , J_5)$	r_{11}
$E_{1G1}(S_1(M_{12}))$	$E_{1G1}(Y_2 , J_2)$	r_{12}
$E_{1G2}(S_1(M_{11}))$	$E_{1G2}(Y_1 , J_5)$	r_{13}
$E_{1G2}(S_1(M_{12}))$	$E_{1G2}(Y_2 , J_2)$	r_{14}
$E_{1G2}(S_1(M_{12}))$	$E_{1G2}(Y_2 , J_3)$	r_{15}
$E_{1G3}(S_1(M_{13}))$	$E_{1G3}(Y_3 , J_8)$	r_{16}
$E_{1G2}(S_1(M_{15}))$	$E_{1G2}(Y_5 , J_3)$	r_{17}
…	…	…
$E_{1Gm}(S_1(M_{1p}))$	$E_{1Gm}(Y_p , J_o)$	r_{1z}

Weder die Policy, noch die eigentlichen Daten sind für T lesbar. Schließlich verfügt er nicht über das Gruppen-One-Time-Pad OTP_{iGj}, das es ihm ermöglichen würde, den Datenschlüssel K_{iGj} aufzudecken. Nur der Zertifizierer verfügt über OTP_{iGj}, jedoch nicht über die Schlüssel- und Datenpakete. Es wird deutlich, dass Zertifizierer und Aufbewahrer nicht böswillig kooperieren dürfen. Dann ändert sich am Schutz der Daten vor einem Missbrauch durch die Aufbewahrungsstelle gegenüber Szenario 1 nichts. Auch Z kann keine Kenntnis der übermittelten Inhalte erlangen, da er nicht über die Datenpakete verfügt.

5.2.3.2 Beitritt zur Empfängergruppe

Fand die initiale Datenablage durch den Dateneigner bei der Aufbewahrungsstelle ähnlich zum Vorgehen in Szenario 1 statt, sind die Schritte dieses Abschnitts ausschließlich für Szenario 2 relevant. Bevor ein potentieller Datenverarbeiter die Übermittlung von Existenznachweisen oder Datenpaketen anfordern kann, muss er zunächst in den Kreis derer aufgenommen werden, denen ein grundsätzliches legitimes Interesse an diesen Informationen zugestanden wird. Das bedeutet, dass sie Mitglied einer Empfängergruppe werden müssen, die bei der Zertifizierungsinstanz geführt wird und die von den Dateneignern als rechtmäßige Menge von Verarbeitern akzeptiert wurde.

2.1 Der Interessent B_j wendet sich mit einer Nachricht an die Zertifizierungsinstanz und beantragt die Aufnahme in eine der verfügbaren Empfängergruppen B_{Gj}. Der Nachricht fügt er die notwendigen Nachweise bei, die ihn als Berechtigten kennzeichnen. Er signiert die Nachricht zudem digital, um deren Herkunft zu belegen.

$$(S_j(\text{„Aufnahme“}, B_j, B_{Gj}, \text{<Nachweise>}))$$

2.2 Der Zertifizierer Z prüft die Signatur und die beigefügten Nachweise:

$$V_j(S_j(\text{„Aufnahme“}, B_j, B_{Gj}, \text{<Nachweise>}))$$

Sofern die Nachweise den Anforderungen für die gewünschte Empfängergruppe genügen, fügt Z den Interessenten als neuen zulässigen Verarbeiter zur Empfängergruppe B_{Gj} hinzu und veröffentlicht diese Änderung an der Gruppenzugehörigkeit[207].

5.2.3.3 Schlüsselabfrage

Wiederum beruhen alle Existenz- und Inhaltsabfragen, die ein Datenverarbeiter B_j an die Aufbewahrungsstelle T richtet, auf dem durch den Dateneigner festgelegten symmetrischen Datenschlüssel.

Der symmetrische Schlüssel K_{iGj} wurde von A_i erstellt und darf nur den Angehörigen der spezifischen Empfängergruppe B_{Gj} bekannt sein. Jeder potentielle Verarbeiter B_j kann die einmalige Schlüsselabfrage zu einem beliebigen Zeitpunkt zwischen der Ablage durch A_i und der ersten Existenz- oder Inhaltsabfrage durch ihn selbst durchführen[208].

3.1 B_j richtet an den Aufbewahrer T eine elektronische Anfrage mit der Kennzeichnung, dass es sich um eine Schlüsselabfrage handelt, sowie seiner eigenen Kennung. T akzeptiert die Anfrage nur, wenn sie durch B_j signiert ist:

$$S_j(\text{„Schlüsselabfrage“}, B_j)$$

3.2 T prüft zunächst, zu welcher Empfängergruppe der Verarbeiter B_j zugeordnet ist. Dazu nutzt er die vom Zertifizierer veröffentlichten Mitgliederlisten der Empfängergruppen.

3.3 Falls er keine entsprechende Zuordnung findet, oder kein Schlüsselpaket für diese Gruppe hinterlegt ist, weist er die Anfrage zurück.

---

[207] Die Veröffentlichung macht natürlich nur unter der zuvor geschilderten Maßgabe Sinn, dass alle Dateneigner identische Zusammensetzungen der Empfängergruppen akzeptieren. Wäre dies nicht der Fall, müsste man die Veröffentlichung durch eine individuelle Information an diejenigen Dateneigner ersetzen, für die sich Empfängergruppen-Änderungen ergeben. Das wiederum würde eine Auflösung der Anonymität aller Dateneigner gegenüber Aufbewahrer und Zertifizierer bedeuten.

[208] Alternativ könnte auch der Zertifizierer Z nach der Aufnahme eines Datenverarbeiters in eine der Empfängergruppen den Schlüsseltransfer zwischen T und B_j initiieren. Dies gäbe ihm jedoch die zusätzliche Möglichkeit, durch entsprechendes Timing den Datenverkehr zwischen beiden abzufangen. Auch wenn Z keinen unmittelbaren Nutzen aus dem Mithören der verschlüsselten Kommunikation hat, könnte er die Kenntnis des Zeitpunkts für Man-in-the-Middle-Angriffe nutzen, falls er böswillige Absichten hegt.

Im Erfolgsfall soll der entsprechende symmetrische Schlüssel K_{iGj} an B_j übermittelt werden. Zur Erinnerung: K_{iGj} liegt beim Aufbewahrer und ist mit einem One-Time-Pad verschlüsselt, das nur Dateneigner und Zertifizierer kennen. Der entsprechende Datensatz bei T hat das Format:

Verarbeiter-gruppe	Datenschlüssel verschlüsselt
$\mathbf{B_{Gj}}$	$\mathbf{OTP_{iGj}(S_i(K_{iGj}))}$

3.3.1 T erstellt zunächst One-Time-Pads OTP_{Tj} („Empfänger-OTP"), die je die Länge der signierten Datenschlüssel aufweisen.

3.3.2 Er verschlüsselt OTP_{Tj} mit dem öffentlichen Schlüssel des Verarbeiters B_j (also mit dem Schlüssel $K_{pub\ j}$) und sendet diese $E_j(OTP_{Tj})$ an B_j. Der Verarbeiter wendet seinen privaten Schlüssel $K_{priv\ j}$ auf die erhaltene Sendung an und legt die ermittelten One-Time-Pads in seiner lokalen Schlüsselverwaltung ab.

3.3.3 Nun verschlüsselt T die für die Gruppe des Verarbeiters vorliegenden Datenschlüssel-Sätze mit den von ihm erzeugten One-Time-Pads durch bitweises Exklusiv-ODER. Er erzeugt also:

$$OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))$$

Das so entstehende Chiffrat ist durch keine der beteiligten Parteien mehr alleine zu entschlüsseln. Über den einen Teil des Geheimnisses, nämlich OTP_{iGj} verfügen nur A_i und Z, den anderen Teil OTP_{Tj} kennen nur T und B_j.

3.3.4 Dieses Chiffrat sendet der Aufbewahrer an den Zertifizierer, der nun den ersten Teil der Verschlüsselung durch erneute XOR-Anwendung des OTP_{iGj}, den A_i ihm ja anfangs geschickt hatte, entfernen kann[209].

Da, wie in Kapitel 6.1 ausgeführt, das Exklusiv-ODER kommutative Schlüsselanwendung erlaubt und gleichzeitig die doppelte Anwendung eines One-Time-Pads auf einen Klartext wieder den Klartext selbst ergibt (d.h. $OTP_1(OTP_1(M)) = M$), gilt:

$$OTP_{iGj}(OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))) =$$

$$= OTP_{Tj}(OTP_{iGj}(OTP_{iGj}(S_i(K_{iGj})))) =$$

[209] Dabei muss der Zertifizierer für das nun übergebene Schlüsselpaket erkennen können, welches der initial übermittelten Gruppen-OTPs das passende ist. In der praktischen Implementierung ist es somit erforderlich, dass die Dateneigner jedes ihrer Schlüsselpakete mit einem eindeutigen, nicht personenbeziehbaren Identifikator versehen, den sie auch dem korrespondierenden Gruppen-OTP beilegen.

$$= OTP_{Tj}(S_i(K_{iGj}))$$

Der Zertifizierer erhält auf diese Weise also den signierten Datenschlüssel, der mit dem One-Time-Pad verschlüsselt ist, das nur dem Aufbewahrer und dem Datenverarbeiter bekannt ist. Für den Zertifizierer selbst ist das Resultat inhaltlich wertlos, da er OTP_{Tj} nicht kennt.

3.3.5 Schließlich übermittelt der Zertifizierer das eben erstellte Chiffrat an den Empfänger B_j. Dieser verfügt über das One-Time-Pad OTP_{Tj} und nutzt es zur Entschlüsselung des Datenschlüssels[210]:

$$OTP_{Tj}(OTP_{Tj}(S_i(K_{iGj}))) = S_i(K_{iGj})$$

Er prüft die Herkunft des Datenschlüssels durch die Validierung der Signatur:

$$V_i(S_i(K_{iGj})) = K_{iGj}$$

3.4 B_j legt den ermittelten Datenschlüssel in einem nur ihm zugänglichen Schlüsselspeicher ab, von wo er ihn für die künftige Nutzung bei allen Anfragen, die sich auf A_i beziehen, abruft.

Auch hier gilt, dass die Schlüsselabfrage durch Mitglieder einer Empfängergruppe immer die Übermittlung aller für diese Gruppe hinterlegten Schlüsselpakete nach sich zieht. Der Aufbewahrer kann nicht selektieren, da er keine Kenntnis hat, welches Schlüsselpaket von welchem Dateneigner stammt.

Abbildung 16 zeigt den Wissensstand, den jede Partei zum Ende der erfolgreichen Schlüsselabfrage hat. Dateneigner und Zertifizierer teilen ein Geheimnis in Form eines gemeinsamen One-Time-Pad. Ebenso nutzen Aufbewahrer und Empfänger ein gemeinsames One-Time-Pad. Durch die wechselweise Anwendung derselben wird der Datenschlüssel K_{iGj} vom Dateneigner zu den Mitgliedern der Empfängergruppe transportiert, ohne dass die Zwischenstationen den Schlüssel entziffern können. Wichtig dabei bleibt, dass Aufbewahrer und Zertifizierer sich nicht bezüglich der ihnen jeweils bekannten One-Time-Pads austauschen und dass alle Kommunikation zusätzlich verschlüsselt ist.

[210] Geht man davon aus, dass ein Verarbeiter Schlüsselpakete für mehr als einen Dateneigner abruft, muss er erkennen können, welches Schlüsselpaket welchem Dateneigner zuzuordnen ist. Dies gelingt, indem er seiner Schlüsselabfrage einen von ihm selbst signierten Identifikator (ID) mitgibt. Aufbewahrer und Zertifizierer reichen dann die ID durch alle folgenden Schritte weiter. Das heißt, der Aufbewahrer übermittelt sie dem Verarbeiter zusammen mit dem Empfänger-OTP und der Zertifizierer versieht das chiffrierte Schlüsselpaket mit ihr. Anschließend kann der Verarbeiter beide Rückmeldungen über die ID mit seiner Abfrage und folglich mit dem Dateneigner assoziieren. Hat er dabei die ID so gewählt, dass sie keine Rückschlüsse auf die Identität des Dateneigners zulässt, bleibt dessen Anonymität gegenüber den anderen Parteien gewahrt.

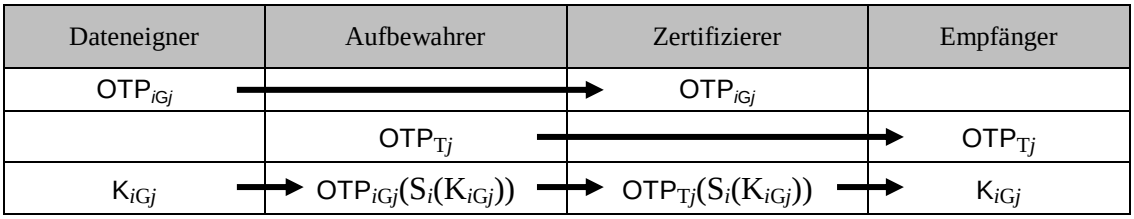


Abbildung 16: Wissensstand nach der Schlüsselabfrage

5.2.3.4 Existenzanfrage

Der Datenverarbeiter kennt nun den Datenschlüssel, den der Eigner für seine Empfängergruppe erstellt hat. Mit diesem Wissen, verbunden mit seiner offiziellen Registrierung als Angehöriger der Empfängergruppe kann er nun die Anfragen analog zu Szenario 1 stellen.

Mit der Existenzanfrage identifiziert der Verarbeiter, ob ein von ihm potentiell benötigter Datensatz zu einer bestimmten Policy vom entsprechenden Eigner hinterlegt ist. Dabei ruft er jedoch nicht die personenbezogenen Daten selbst ab.

4.1 B_j bestimmt die Kombination von Datentyp Y_l und dem angestrebtem Verwendungszweck J_k.

4.2 Diese verschlüsselt er mit dem Datenschlüssel K_{iGj}: $E_{iGj}(Y_l, J_k)$. Dadurch erstellt er einen Vergleichswert $P_{iGjlk}´$ für die von A_i hinterlegte Policy P_{iGjlk}.

4.3 B_j übermittelt dieses Chiffrat zusammen mit der Kennzeichnung, dass es sich um eine Existenzanfrage handelt, und von ihm signiert, an T.

$$S_j(\text{„Existenzanfrage“}, P_{iGjlk}´)$$

4.4 T überprüft die Signatur sowie die Zugehörigkeit des B_j zu einer gültigen Empfängergruppe, wie sie von Z veröffentlicht sind. Er vergleicht anschließend das übersandte $P_{iGjlk}´$ mit allen Einträgen P_{iGjlk} seiner Datenbank.

T kennt nach wie vor nicht den Schlüssel K_{iGj}, daher kann er aus der Anfrage keine weiteren Schlüsse auf Art oder Inhalt der Daten ziehen, für die sich B_j interessiert. Er weiß nur, dass die Anfrage von B_j kommt. Auch kann er aus der Anfrage nicht ermitteln, von wem die Policy und der dazu hinterlegte Datensatz stammen[211].

4.5 T übermittelt das Ergebnis des Vergleichs an B_j. Findet er einen Eintrag, für den gilt $P_{iGjlk} = P_{iGjlk}´$, übermittelt er „Ja“, anderenfalls „Nein“. Der Verarbeiter weiß

[211] Gleiches gilt im Übrigen auch für den Zertifizierer, der aber ohnehin nach einer erfolgreichen Schlüsselabfrage in die weitere Kommunikation nicht mehr eingebunden ist.

nun, ob A_i Daten für seine Empfängergruppe hinterlegt hat, die der angefragten Kombination aus Datentyp und Verarbeitungszweck entsprechen.

4.6 T versieht die Existenzanfrage mit einem Zeitstempel und speichert sie zusammen mit der Antwort in seiner Protokolldatei.

5.2.3.5 Inhaltsabfrage

Auch die Inhaltsabfrage verläuft analog zu Szenario 1.

5.1 B_j bestimmt zunächst die Kombination von Datentyp Y_l und von ihm angestrebtem Verwendungszweck J_k.

5.2 Diese verschlüsselt er mit dem Datenschlüssel K_{iGj} zu $E_{iGj}(Y_l, J_k)$ und erhält analog zur Existenzanfrage den Policy-Vergleichswert $P_{iGjlk}´$.

5.3 B_j übermittelt $P_{iGjlk}´$ zusammen mit der Kennzeichnung, dass es sich um eine Inhaltsabfrage handelt, und von ihm signiert, an T.

$$S_j(\text{„Inhaltsabfrage“}, P_{iGjlk}´)$$

5.4 T überprüft die Signatur sowie die Zugehörigkeit des B_j zu einer gültigen Empfängergruppe, wie sie durch Z veröffentlicht sind. Anschließend vergleicht er das übermittelte $P_{iGjlk}´$ mit allen Einträgen P_{iGjlk} seiner Datenbank.

5.5 Hat T einen Datensatz mit $P_{iGjlk} = P_{iGjlk}´$ gefunden, übermittelt er das korrespondierende Datenpaket $E_{iGj}(S_i(M_{il}))$ an B_j. Existiert kein korrespondierendes Paket zur Policy $P_{iGjlk}´$ erhält der Verarbeiter analog zur Existenzanfrage ein „Nein“ zur Antwort.

5.6 B_j kann im Erfolgsfall das erhaltene Paket mit Hilfe des Schlüssels K_{iGj} in den Klartext umsetzen und die Signatur von A_i mit dessen öffentlichem Schlüssel prüfen:

$$M_{il} = V_i(D_{iGj}(E_{iGj}(S_i(M_{il}))))$$

5.7 T erstellt über die Herausgabe des Datensatzes einen Logeintrag, den er zusammen mit der zugehörigen Anfrage durch B_j und einem Zeitstempel speichert.

5.2.3.6 Logabfrage

Dateneigner A_i kann das Protokoll der für seine Datensätze gestellten Schlüssel-, Existenz- und Inhaltsabfragen vom Aufbewahrer abrufen, ohne seine Identität preiszugeben.

6.1 A_i wählt aus den von ihm initial erstellten verschlüsselten Policies P_{iGjlk} diejenigen aus, für die er die Anfrageprotokolle abrufen möchte.

6.2 Er übermittelt die Liste dieser Policies $P_{iGjlk}´´$ mit der Kennzeichnung, dass es sich um eine Logabfrage handelt an T.

6.3 Für jede Logabfrage mit einer übermittelten Policy $P_{iGjlk}´´$ durchsucht T seine Logtabelle und erstellt eine Liste aller Existenz- und Inhaltsabfragen, für die $P_{iGjlk}´´ = P_{iGjlk}$.

6.4 Er stellt A_i die entstandene Liste zur Verfügung. A_i kann nun jede Anfrage, die sich auf seine Daten bezieht, mit dem entsprechenden Zeitstempel überprüfen. Die Verarbeiter können nicht abstreiten, die Anfragen gestellt zu haben, da diese mit ihrer persönlichen digitalen Signatur versehen sind.

Für das Szenario 2 ist es von besonderer Bedeutung, dass die digitalen Signaturen jeweils gemeinsam mit den Anfragen im Protokoll gespeichert werden. In Szenario 1 konnte man noch aus der Tatsache, dass ein Verarbeiter sinnvoll verschlüsselte Anfragen $P_{ijlk}´$ stellt, darauf schließen, dass er den exklusiv für ihn erstellten Datenschlüssel K_{ij} kennt, was zu einem Grad seine Identität als Anfragender nachweist. In Szenario 2 haben potentiell alle Mitglieder der Empfängergruppe Zugriff auf den Datenschlüssel K_{iGj}, womit auch die Anfragen $P_{iGjlk}´$ nur auf die Gruppenmitgliedschaft, nicht aber auf das Individuum verweisen. Die individuelle Signatur ist daher notwendig, um eindeutig den Urheber jeder Anfrage im Log darzustellen.

5.2.3.7 Datenänderungen durch den Eigner

Die folgenden Änderungen erfolgen analog zu Szenario 1 und werden hier nicht mehr detailliert dargestellt:

7.1 Generelles Ausscheiden des Dateneigners aus dem System: A_i sendet Löschanforderungen für alle von ihm abgelegten Datensätze an den Aufbewahrer, wie im nächsten Abschnitt beschrieben[212]. Ebenso kann er Löschanforderungen für die hinterlegten Schlüsselpakete und One-Time-Pads an Aufbewahrer und Zertifizierer senden[213].

7.2 Entfernen von Inhalten: Möchte A_i einen Datensatz bei T löschen, sendet er die durch die Policy spezifizierte Löschaufforderung anonym an T und weist seine

[212] Dieses Vorgehen muss der Dateneigner im Übrigen auch wählen, falls er das Vertrauen in den Zertifizierer verloren hat, beispielsweise wenn dieser zweifelhafte Mitglieder in die Empfängergruppen aufgenommen hat. Er könnte im Anschluss an sein Ausscheiden mit seinen Datensätzen in ein anderes System umsteigen, möglicherweise unter Beibehaltung der etablierten Beziehung zu seinem bisherigen Aufbewahrer.

[213] Dies erfolgt unter der Prämisse anonym, dass auch diese Pakete analog zu den Datenpaketen mit einer Zufallszahl versehen wurden, deren Kenntnis die Eignerschaft über den Eintrag beweist.

Eignerschaft der Datensätze dadurch nach, dass er den Hashwert der Zufallszahlen r_{ix} mitliefert:

$$E_T(\text{„Löschen“}, P_{iGjlk}, h(r_{ix}))$$

T ermittelt seinerseits $r_{ix}' = h(r_{ix}))$ für das r_{ix}, das durch die initiale Datenablage dem über P_{iGjlk} identifizierten Datensatz zugeordnet ist. Wenn der von ihm berechnete Wert mit dem übermittelten Wert übereinstimmt, löscht er den entsprechenden Datensatz aus seiner Datenbank.

7.3 Hinzufügen einer Empfängergruppe und entsprechender Daten: Das Vorgehen ist analog zur initialen Datenablage. A_i sendet getrennt voneinander ein zusätzliches Schlüsselpaket und korrespondierende Datenpakete an T, sowie zusätzliche One-Time-Pads an Z.

7.4 Hinzufügen von Inhalten (für bereits etablierte Empfängergruppen): A_i sendet anonym zusätzliche Datenpakete an T.

7.5 Ausschließen einer Empfängergruppe: A_i lässt alle für B_{Gj} hinterlegten Einträge im Datenpaket löschen, wie in „Entfernen von Inhalten“ beschrieben.

7.6 Änderung bestehenden Inhalts: Analog zur Löschung dient die initial mit den Datensätzen assoziierte Zufallszahl r_{ix} als Beweis, dass ein Datensatz von demjenigen stammt, der nun eine Aktualisierung veranlassen möchte. Wieder übermittelt der Dateneigner die Policy als Identifikator und den Hashwert über das korrespondierende r_{ix}. Im Gegensatz zur Löschanforderung nimmt er nun auch zusätzlich den veränderten Datensatz $E_{iGj}(S_i(M_{il}))'$ auf, der anstelle des bisherigen Datensatzes zur Policy P_{iGjlk} hinterlegt werden soll. Die Nachricht lautet:

$$E_T(\text{„Änderung“}, P_{iGjlk}, E_{iGj}(S_i(M_{il}))', h(r_{ix}))$$

Anschließend ersetzen sowohl A_i als auch T in ihren Datentabellen den in der geänderten Zeile gespeicherten Wert r_{ix} durch $h(r_{ix})$. Für alle künftigen Änderungsanforderungen gilt dann $r_{ix}' = h(r_{ix})$. So werden Replay-Angriffe verhindert.

7.7 Änderung der Zuweisung von Datentyp oder Verarbeitungszweck: Diese Änderungen sind wie die Löschung eines Datensatzes und die Lieferung eines neuen Pakets zu behandeln.

7.8 Aktualisierung von alten Existenzanfragen nach der Löschung von Datensätzen: T prüft alle historischen Existenzanfragen aus seinem Protokoll erneut, wenn eine Löschanforderung durch den Dateneigner initiiert wurde. Dazu vergleicht er die von A_i als Identifikator übermittelten P_{iGjlk} mit den im Protokoll als abgefragt gekennzeichneten Policies und informiert bei Übereinstimmung die Mitglieder der jeweiligen Empfängergruppe B_{Gj}, die bereits Existenzanfragen gestellt hatten.

5.2.3.8 Änderungen durch den Zertifizierer

Es liegt in Szenario 2 nicht mehr in der Zuständigkeit des Dateneigners, spezifische Empfänger für seine Daten auszuwählen. Vielmehr legt er die für ihn relevanten Empfängergruppen fest und muss es dem Zertifizierer überlassen, die konkreten Gruppenmitglieder zu prüfen und zuzulassen.

8.1 Aufnahme eines neuen Gruppenmitglieds in eine bestehende Empfängergruppe: Das Hinzufügen eines neuen Mitglieds kann zu jedem Zeitpunkt – vor oder nach der Datenablage durch die Dateneigner – erfolgen und folgt immer den in Kapitel 5.2.3.2 beschriebenen Schritten.

8.2 Entfernen eines Mitglieds aus einer Empfängergruppe: Bevor Datenverarbeiter eine Schlüsselabfrage durchgeführt haben, verfügen sie noch über keinerlei zusätzliche Informationen. Entsprechend einfach ist ein Entfernen aus der Empfängergruppe. Z muss lediglich die Veröffentlichung der Gruppenzugehörigkeit zurücknehmen. Sollte nun der entfernte Verarbeiter dennoch eine Schlüsselabfrage an T stellen, würde dieser zunächst in den von Z bereitgestellten Mitgliederlisten nachschlagen und das Protokoll abbrechen, da er dort keinen korrespondierenden Eintrag findet.

Auch wenn das Entfernen aus der Empfängergruppe erst nach erfolgter Schlüsselübertragung stattfindet, nimmt Z natürlich die Veröffentlichung des betreffenden Verarbeiters als Mitglied einer der Empfängergruppen zurück. Da der Verarbeiter aber bereits über den Datenschlüssel für seine (ehemalige) Gruppe verfügt, kann er gültige Existenz- und Inhaltsabfragen an T stellen. T überprüft jedoch vor einer Rückmeldung, ob der Verarbeiter, der die Anfrage signiert hat, in den Empfängergruppen-Listen des Zertifizierers vorliegt. So erkennt er die Änderung und reagiert nicht mehr auf die Anfragen. Es spielt für Z also für das Entfernen von Verarbeitern aus Empfängergruppen keine Rolle, ob diese bereits Schlüsselabfragen durchgeführt haben.

Haben bereits Existenz- oder Inhaltsabfragen stattgefunden, ändert auch das nichts an der Situation. Die übermittelten Daten können ohnehin nicht zurückgeholt werden und die Abfrage neuer Inhalte wird wiederum durch die Prüfung der veröffentlichten Mitgliederlisten durch T verhindert.

8.3 Aufspalten einer Empfängergruppe in mehrere Gruppen: Es kann geschehen, dass eine etablierte Empfängergruppe in mehrere neue Gruppen aufgeteilt werden muss. Dies kann rechtliche oder organisatorische Gründe haben. So könnte es beispielsweise sein, dass die Gruppe der teilnehmenden Notfall-Ärzte so groß wird, dass sie nach Regionen oder Spezialrichtungen aufgeteilt werden sollen.

Soll dies mit einer vollständigen Auflösung der bestehenden Gruppen einhergehen, so ist das Protokoll zur Bildung neuer Gruppen von Beginn an neu zu durchlaufen. Damit geht auch die Neugenerierung aller Datenschlüssel und One-Time-Pads

einher. Die bisher beim Aufbewahrer für die Empfängergruppe hinterlegten Schlüsselpakete und Datenpakete müssen hinsichtlich der neuen Datenschlüssel und mit neuer Aufteilung auf die Empfängergruppen von den Dateneignern neu erstellt werden. Alle Resultate der durch Mitglieder der betroffenen Empfängergruppe durchgeführten Existenzanfragen sind hinfällig.

Eine andere Variante ist die Aufspaltung in der Form, dass von einer existierenden Empfängergruppe Teile der Empfänger in eine neue Gruppe verlegt werden. In diesem Fall sollen die etablierten Beziehungen zu den in der Ursprungs-Gruppe verbleibenden Mitgliedern, das heißt die ausgetauschten Schlüssel und die bereits erfolgten Existenzanfragen, ihre Gültigkeit behalten.

Die zu verlagernden Mitglieder verfügen potentiell über den bislang gültigen Gruppen-Datenschlüssel. Jeder Empfänger könnte sich natürlich den Schlüssel zur Sicherheit kopiert haben, so dass es nicht ausreichend wäre, der Schlüsselverwaltung des Empfängers einen Befehl zum Löschen des bisher bekannten Datenschlüssels zu senden. Vielmehr muss dafür gesorgt werden, dass dieser Datenschlüssel nicht für weitere Anfragen genutzt werden kann. Hier findet analog das Vorgehen zum Entfernen eines Mitglieds aus einer Empfängergruppe, wie oben beschrieben, Anwendung. Das bedeutet, der Aufbewahrer beantwortet keine empfängergruppenspezifischen Anfragen eines Mitglieds, das nicht mehr in der vom Zertifizierer veröffentlichten Mitgliedsliste derjenigen Gruppe geführt ist, für die es die Anfrage stellt.

Im Gegenzug sind jedoch hier die Dateneigner zu involvieren. Sie werden durch den Zertifizierer über die neue Empfängergruppe informiert und erstellen neue Policies, Daten- und Schlüsselpakete für diese Gruppe, gemäß ihrer Einschätzungen des Datenbedarfs.

5.2.4 *Exkurs*: Rückführung auf Szenario 1

In Schritt 1.8 zur Datenablage in Szenario 2 hat der Dateneigner die One-Time-Pads mit dem öffentlichen Schlüssel des Zertifizierers chiffriert, bevor er sie diesem übermittelt hat. Warum aber hat Z nicht einfach ein zusätzliches Schlüsselpaar je Empfängergruppe erzeugt, und den Eigner dazu aufgefordert, die Datenpakete und Policies mit den öffentlichen Schlüsseln dieser Schlüsselpaare anstelle des öffentlichen Schlüssels von Z zu chiffrieren? In diesem Fall müsste Z einem neuen Gruppenmitglied nach dessen Aufnahme nur den privaten Teil des entsprechenden Schlüsselpaars aushändigen. Der Verarbeiter würde diesen Schlüssel nutzen, um sich das Schlüsselpaket zu besorgen, es zu dechiffrieren und fortan analog zu Szenario 1 Existenz- und Inhaltsabfragen an den Aufbewahrer stellen (dann jedoch unter Nutzung seines eigenen privaten Schlüssels und nicht des privaten Empfängergruppen-Schlüssels; dies wäre notwendig, damit die entstehenden Protokolleinträge beim Aufbewahrer später dem individuellen Verarbeiter und nicht nur der Gruppe zugeordnet werden können).

Ein Verarbeiter, dem der private Schlüssel der Empfängergruppe bekannt ist, kann alle Datensätze entschlüsseln, die für diese Empfängergruppe vorgesehen sind. Dies entspräche Szenario 1, auch hier unterläge die Aufsicht über die Auslieferung der verschlüsselten Datensätze wieder alleine dem Aufbewahrer T.

Dennoch wären mit der Herausgabe eines privaten Schlüssels der Empfängergruppe an Gruppenmitglieder zusätzliche Risiken verbunden:

Offensichtlich sind die Datenschlüssel das kritische Element im Protokoll. Wer über sie verfügt, kann die für Anfragen benötigten Vergleichspolicies erstellen und kann die Datenpakete entschlüsseln. Die Datenschlüssel sind also mit höchster Priorität zu schützen, es ist zu verhindern, dass sie mehr Parteien als den ursprünglich beabsichtigten bekannt werden. Im Protokoll mit der kommutativen Anwendung der OTPs beim Schlüsselaustausch verfügen T und Z zu keinem Zeitpunkt über hinreichende Informationen, um den Datenschlüssel im Klartext identifizieren zu können (vorausgesetzt, sie kooperieren nicht in der Form, dass sie Ihre OTPs untereinander weitergeben).

Würde man jedoch mit privaten Schlüsseln für die Empfängergruppen oder mit von A_i auf Vorrat generierten individuellen Empfängerschlüsseln arbeiten, würde für Z das Abhören einer unverschlüsselten Kommunikation zwischen T und einem Verarbeiter genügen, und er hätte alle Informationen, die er benötigt, um K_{iGj} zu entschlüsseln, was das System kompromittieren würde.

Würde einer der bei Z lagernden privaten Schlüssel der Empfänger(gruppen) bei T bekannt, könnte wiederum dieser die Dechiffrierung der entsprechenden Datenschlüssel vornehmen, was das System ebenfalls kompromittieren würde.

Die beiden eben geschilderten Angriffe überschneiden sich mit den Implikationen des Angreifermodells „T und Z kooperieren“. Sofern dies eintritt, ist das System ohnehin kompromittiert. Die beiden genannten Angriffe können jedoch auch unabhängig von einer Kooperation erfolgen. Dies bedeutet im Rückschluss, dass die größere Sicherheit durch ein System erreicht wird, das nur durch Kooperation ausgehebelt werden kann. Dies ist bei der wechselseitigen Anwendung der OTPs zum Schlüsselaustausch gegeben, so dass diese Variante sicherer als die eigene Erzeugung von Empfänger(gruppen)schlüsseln durch Z zu werten ist.

6 Überlegungen zur Implementierung

6.1 Kryptoalgorithmen

6.1.1 Sicherung des Schlüsselaustauschs

Für den kryptographischen Algorithmus zum Austausch des Datenschlüssels in Szenario 2 besteht neben den allgemeinen Anforderungen an ein sicheres Kryptosystem im Besonderen die folgende Forderung: Der Algorithmus muss in der Anwendung von mehreren Schlüsseln bzw. Schlüsselpaaren kommutativ sein, das bedeutet:

$$D_2(D_1(E_1(E_2(m)))) = D_2(D_1(E_2(E_1(m)))) = m$$

mit D_1: Dechiffrierung mit dem Schlüssel 1; E_1: Chiffrierung mit dem Schlüssel 1; D_2 und E_2 analog für Schlüsselpaar 2.

Im konkreten Fall des Szenarios 2 müssen die folgenden Schritte in dieser Reihenfolge getätigt werden (die für die Kommutativität irrelevanten Schritte sind hier nicht dargestellt):

1. A_i verschlüsselt den Datenschlüssel so, dass Z ihn später entschlüsseln kann.
2. T erhält den verschlüsselten Datenschlüssel und verschlüsselt ihn erneut so, dass B_j diese zweite Verschlüsselung entschlüsseln kann.
3. T leitet den doppelt verschlüsselten Datenschlüssel an Z. Dieser löst die von A_i angebrachte Verschlüsselung und leitet das Ergebnis an B_j weiter.
4. B_j löst die von T angebrachte Verschlüsselung und erhält den Datenschlüssel im Klartext.

Das Protokoll ist also darauf angewiesen, dass die Anwendung der verschiedenen Schlüssel auf ein mehrfach verschlüsseltes Chiffrat nicht nur nach dem Last-in-First-out Prinzip, sondern auch in anderer Reihenfolge mit dem korrekten Ergebnis erfolgt. Viele Verfahren der Public-Key-Kryptographie wie auch symmetrische Blockchiffrierer erfüllen diese Anforderung nicht. RSA als prominentester Vertreter verfügt allgemein nicht über diese Eigenschaft. Dennoch könnte RSA genutzt werden, falls man sich darauf einigt, dass alle Parteien ihr eigenes Paar von öffentlichem und privatem Schlüssel besitzen, diese aber jeweils über denselben Modul berechnet wurden.

6.1.2 RSA

Sind p und q große Primzahlen und $n = pq$, wählt man für RSA eine Zahl e, die zu $(p-1)(q-1)$ relativ prim ist, und berechnet zum Beispiel über den erweiterten Euklidischen

Algorithmus ein d, für das gilt: $d = e^{-1} \bmod ((p-1)(q-1))$. Unter Kenntnis des öffentlichen RSA-Schlüssels aus e und n können Kommunikationspartner die Verschlüsselung $c = m^e \bmod n$ durchführen, die der Empfänger mit Hilfe des privaten Schlüssels d wieder dechiffrieren kann: $m = c^d \bmod n$ (wobei $m<n$)[214].

Ersetzt man in obiger Ver- und Entschlüsselungsreihenfolge $D_2(D_1(E_2(E_1(m))))$ die Operationen D_1, D_2, E_1 und E_2 durch den jeweiligen Rechenweg für RSA, wobei d_1, d_2, e_1, e_2, n_1 und n_2 die jeweiligen Komponenten e, d, n für Schlüsselpaar 1 bzw. 2 darstellen, erhält man:

$$(((m^{e_1} \bmod n_1)^{e_2} \bmod n_2)^{d_1} \bmod n_1)^{d_2} \bmod n_2$$

Dieses lässt sich nur dann zur Identität mit m umformen, wenn $n_1 = n_2$ ($=n$). Dann gilt zum einen für die Kommutativität:

$$\begin{aligned}
&(((m^{e_1} \bmod n)^{e_2} \bmod n)^{d_1} \bmod n)^{d_2} \bmod n = \\
&= m^{e_1 \cdot e_2 \cdot d_1 \cdot d_2} \bmod n = \\
&= m^{e_1 \cdot e_2 \cdot d_2 \cdot d_1} \bmod n = \\
&= (((m^{e_1} \bmod n)^{e_2} \bmod n)^{d_2} \bmod n)^{d_1} \bmod n \\
&\text{(Kommutativität)}
\end{aligned}$$

Und zum anderen für die De-/Chiffriereigenschaft:

$$\begin{aligned}
&m^{e_1 \cdot e_2 \cdot d_2 \cdot d_1} \bmod n = \\
&= m^{e_1 \cdot e_2 \cdot (1/e_2) \cdot (1/e_1)} \bmod n = \\
&(d \text{ ist definiert als Kehrwert von } e \bmod n) \\
&= m^{(e_1/e_1) \cdot (e_2/e_2)} \bmod n = m^1 \bmod n = m \\
&(\text{da } m<n).
\end{aligned}$$

Würde also Z ein gemeinsames n für A_i und eine bestimmte Empfängergruppe B_{Gj} festlegen, könnten für diese jeweils eigene RSA-Schlüsselpaare generiert werden und sie könnten am kommutativen Protokoll mit ihren Schlüsselpaaren teilnehmen.

Leider ist RSA beim gemeinsamen Einsatz von Schlüsselpaaren, die über denselben Modul erzeugt wurden, sehr einfach angreifbar. Simmons zeigt, wie ein Kryptanalytiker aus der Kenntnis von zwei Chiffretexten, den beiden öffentlichen Schlüsseln sowie dem gemeinsamen Modul n auf einfache Weise den Klartext ermitteln kann[215]. DeLaurentis entwirft zudem einen Angriff, durch den n faktorisiert wird, was einer Kompromittierung des Systems entspricht[216].

[214] vgl. [RSA78]
[215] vgl. [Simm83]
[216] vgl. [DeLa84]

Schließlich kommt Moore für RSA unter anderem zu den folgenden Einschränkungen[217]:

- Ein Angreifer, der ein Verschlüsselungs-/Entschlüsselungspaar von Exponenten für einen gegebenen Modul kennt, kann den Modul faktorisieren. Er benötigt lediglich ein e_1 und ein e_2 (mit $e_1 \neq e_2$) als gültige Verschlüsselungsexponenten zum gemeinsamen Modul n.
- Ein Angreifer, der ein Verschlüsselungs-/Entschlüsselungspaar von Exponenten für einen gegebenen Modul kennt, kann andere Verschlüsselungs-/Entschlüsselungspaare berechnen, ohne n zu faktorisieren.

Aus diesen beiden Punkten folgt:

- Ein Protokoll, das RSA in einem Kommunikationsnetz einsetzt, darf keinen gemeinsamen Modul verwenden.

Im Ergebnis ist also zusammenzufassen, dass RSA für Szenario 2 nicht geeignet ist.

6.1.3 Diffie-Hellmann

Schlüsselaustauschverfahren wie Diffie-Hellmann[218] könnten ebenfalls geeignet sein, geht es doch darum, eine Verschlüsselung zu erzielen, die zwar von Dateneigner und Empfänger dechiffriert werden kann, nicht jedoch von Aufbewahrungsstelle und Zertifizierer.

Als Hindernis erweist sich hier jedoch die zeitliche Abfolge der Schritte. Bevor A_i sein Datenpaket an T weitergibt, muss er es verschlüsseln. Mit B_j kann er jedoch zu diesem Zeitpunkt keinen Schlüssel austauschen, schließlich ist der Empfänger hier namentlich noch nicht bekannt.

Für den Diffie-Hellmann-Schlüsselaustausch vereinbaren beide Parteien eine Zahl g und eine Primzahl n als Modul über einen öffentlichen Kanal. Beide Parteien wählen geheim je eine Zahl, x und y und berechnen $X = g^x \bmod n$ bzw. $Y = g^y \bmod n$. Sie tauschen X und Y aus und berechnen über ihre Geheimnisse x und y den Schlüssel $k = Y^x \bmod n$ bzw. $k = X^y \bmod n$.

Es wird deutlich, dass ein Kommunikationsteilnehmer erst dann k ermitteln und mit diesem eine Nachricht verschlüsseln kann, wenn er das Ergebnis der Berechnung beim jeweils anderen, also X und Y erhalten hat. A_i kann also keinen Schlüssel festlegen, solange B_j noch nicht am Protokoll teilnimmt. B_j wird bei PDG/v zum Zeitpunkt der Datenhinterlegung von Z vertreten. Wenn Z jedoch anstelle des B_j an der Schlüsselerzeugung mitwirkt, hat er auch Zugang zu den verschlüsselten Daten. Daher ist der Diffie-Hellmann-Schlüsselaustausch in dieser Form für das Protokoll nicht geeignet.

[217] vgl. [Moor92]
[218] vgl. [DiHe76]

Eine von Hughes 1994 vorgeschlagene Variante kommt der Lösung näher[219]. A_i wählt hierbei alleine den Modul *n*, *g* (die er veröffentlicht) und sein Geheimnis *x*. Er errechnet den Schlüssel $k = g^x \bmod n$ und kann *k* bereits zur Verschlüsselung seiner Daten nutzen, ohne mit einer anderen Partei interagiert zu haben.

Will B_j später den Schlüssel *k* ermitteln, wird er jedoch basierend auf seinem Geheimnis *y* Folgendes berechnen: $Y = g^y \bmod n$, das ermittelte *Y* an A_i senden und aus dessen Antwort $X = Y^x \bmod n$ den Schlüssel *k* ermitteln:

$$z = y^{-1} \bmod (n\text{-}1)$$

und dann

$$k = X^z \bmod n$$

Damit erfordert dieses Protokoll, dass A_i zum Zeitpunkt des Datenabrufs verfügbar sein muss, was der PDG-Anforderung III aus Kapitel 4.2 zuwiderläuft.

6.1.4 One-Time-Pad

Das One-Time-Pad mit bitweisem Exklusiv-ODER (XOR) als mathematisch nachweisbar nicht zu brechende und zugleich denkbar einfache Verschlüsselung[220], scheint geeignet:

- Der signierte Schlüssel $S_i(K_{iGj})$ ist von überschaubarer und vor allem konstanter Größe – vorausgesetzt man einigt sich auf einen Signieralgorithmus mit einheitlicher Schlüssellänge. Da ein One-Time-Pad immer mindestens die Länge des zu verschlüsselnden Klartextes haben muss, ist dies ein wichtiger Faktor für die Handhabbarkeit.
- Die Schlüsselanwendung bei bitweisem XOR ist wie oben gefordert kommutativ.
- Die doppelte XOR-Anwendung eines Schlüssels auf einen Klartext ergibt wieder den Klartext selbst.

Schneier weist jedoch darauf hin, dass die beiden letztgenannten Eigenschaften ein Risiko bei der Anwendung des One-Time-Pad mit XOR darstellen können[221]. Dazu muss man die Optionen betrachten, die ein Angreifer hätte, dem es gelänge, alle verschlüsselten Nachrichten abzuhören.

Im Falle des zuvor geschilderten Protokolls wären dies die Nachrichten:

1. Nachricht von A_i an T zur initialen Hinterlegung des Schlüsselpakets:

$$OTP_{iGj}(S_i(K_{iGj}))$$

[219] vgl. [Schn96] S.588f
[220] vgl. dazu das Plädoyer von Dirk Rijmenants [Rijm10].
[221] vgl. [Schn96] S.591f

2. Nachricht von T an Z mit dem doppelt verschlüsselten Schlüsselpaket:

$$OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))$$

3. Nachricht von T an B_j mit dem teilentschlüsselten Schlüsselpaket:

$$OTP_{Tj}(S_i(K_{iGj}))$$

Verknüpft der Angreifer nun diese Nachrichten seinerseits mit XOR ($\oplus$), erhält er:

$$(OTP_{iGj}(S_i(K_{iGj}))) \oplus (OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))) \oplus (OTP_{Tj}(S_i(K_{iGj})))$$

Da die One-Time-Pad Verschlüsselung $OTP_{xy}(z)$ gleichbedeutend ist mit der XOR-Verknüpfung des Pads mit dem Klartext, also $OTP_{xy}(z) = OTP_{xy} \oplus z$, lässt sich die Verknüpfung durch den Angreifer auch schreiben als:

$$(OTP_{iGj} \oplus S_i(K_{iGj})) \oplus (OTP_{Tj} \oplus (OTP_{iGj} \oplus S_i(K_{iGj}))) \oplus (OTP_{Tj} \oplus S_i(K_{iGj}))$$

Wegen der Kommutativität ist das gleichbedeutend mit:

$$OTP_{iGj} \oplus S_i(K_{iGj}) \oplus OTP_{Tj} \oplus OTP_{iGj} \oplus S_i(K_{iGj}) \oplus OTP_{Tj} \oplus S_i(K_{iGj}) =$$
$$= OTP_{iGj} \oplus OTP_{iGj} \oplus OTP_{Tj} \oplus OTP_{Tj} \oplus S_i(K_{iGj}) \oplus S_i(K_{iGj}) \oplus S_i(K_{iGj}))$$

Eliminiert man nun die jeweils doppelten XOR-Anwendungen von OTP_{iGj}, OTP_{Tj} und $S_i(K_{iGj})$, dann bleibt:

$$S_i(K_{iGj})$$

Damit hätte der Angreifer den signierten Datenschlüssel des Eigners A_i für die Empfängergruppe B_{Gj} im Klartext vorliegen. Hier zeigt sich also, dass die Kommunikation zwischen allen beteiligten Parteien des Protokolls zusätzlich durch Public-Key-Kryptographie geschützt werden muss, und dass das Protokoll somit nicht die informationstheoretische Sicherheit des One-Time-Pad bietet, sondern maximal die komplexitätstheoretische Sicherheit des verwendeten Public Key-Algorithmus[222].

Zentral für das One-Time-Pad ist zudem die Qualität, also die Zufälligkeit, des Schlüssels. One Time Pads können nur dann Sicherheit bieten, wenn ein zuverlässiger Zufallszahlengenerator zur Generierung der Pads zur Verfügung steht. Theoretisch sichere Generatoren sind mit relativ hohem Aufwand zu implementieren. Insbesondere sind sie vom Einsatz spezifischer Hardware abhängig. Die theoretisch perfekte Sicherheit des One Time Pad[223] ist aber ohnehin nicht auf das vorgestellte Verfahren zu übertragen, da die

[222] Alternativ könnte statt des One-Time-Pad das Three-Pass-Protokoll von Shamir Anwendung finden, das sich an RSA orientiert, und das oben beschriebene Risiko durch Kombination mehrerer abgehörter Nachrichten nicht trägt, vgl. [Mass92].

[223] Der Beweis für die informationstheoretische Sicherheit des OTP bei der Nutzung echter Zufallszahlen wird beispielsweise in [TaWe06] geführt: Zur Erfüllung von Shannons Forderung an ein perfekt sicheres Kryptosystem (Ein Angreifer erhält aus dem Kryptogramm keine neue Information) wird nachgewiesen, dass für eine gegebenes Kryptogramm, das mit einem echten OTP verschlüsselt wurde, alle möglichen Ausgangsnachrichten gleich wahrscheinlich sind.

Pads selbst durch asymmetrische Kryptographie und die verschlüsselten Daten mit anderen symmetrischen Verfahren verschlüsselt sind, die zumindest gegenüber Brute-Force Angriffen keine perfekte Sicherheit bieten. Die Zufallszahlen der One Time Pads absolut zufällig und den Einsatz im Verfahren damit theoretisch sicher zu machen, ist verzichtbar. Es genügt, wenn die erzeugten Zufallszahlen praktisch sicher sind, also mit Pseudozufallszahlengeneratoren erzeugt werden, solange diese Sicherheit zu kompromittieren mindestens ebenso aufwendig ist, wie das Brechen der anderen eingesetzten Kryptosysteme[224].

Eine Definition für gute Zufallszahlen und deren Erzeugung liefert RFC 4086[225]. Dessen Autoren zeigen, dass aktuelle Computer über die Hardware-Mittel verfügen, die zur Erzeugung qualitativ hochwertiger Zufallszahlen genutzt werden können, etwa Laufwerke und Mikrofone. Hat man solchermaßen einige Hundert Bits Zufallszahlen generiert, lassen diese sich als Ausgangsbasis („Seed“) für Software-Algorithmen (Pseudozufallszahlen-generatoren) nutzen, die große Mengen von Zahlen mit praktisch zufälliger Verteilung generieren können.

6.2 Architektur für Szenario 1

PDG benötigt neben den Standard-Komponenten zur Datenübermittlung weitere Applikationen und Datenbanken, die beim Betroffenen, bei der Aufbewahrungsstelle und beim Empfänger eingesetzt werden. Eine schematische Darstellung der Komponenten ist in Kapitel 6.2.8 zu finden.

6.2.1 Datenerfassung und persönlicher Datenspeicher

In dieser Komponente erfasst der Betroffene alle Datensätze, die zur Weitergabe vorzusehen sind. Die Erfassung erfolgt gemäß den Feldern und folgt der Nomenklatur, die durch den allgemein verabschiedeten Datentypen-Katalog festgelegt sind.

Die Datensätze werden mit einem Zeitstempel versehen in den persönlichen Datenspeicher abgelegt, und einer Versionskontrolle unterworfen.

Die Kontrolle über diese Komponente liegt vollständig beim Dateneigner, der sie in Form einer lokalen Installation auf einem entsprechend gesicherten Rechner betreibt. Es besteht eine eingehende Schnittstelle, die dem Import der Datentypen-Kataloge dient. Eine ausgehende Schnittstelle beliefert die ebenfalls lokal liegende Daten-/Policy-Verwaltung mit den jeweils benötigten Datensätzen.

[224] vgl. [Rijm10a], ebenso zu Optionen praktisch sicherer Zufallszahlenerzeugung.
[225] http://www.ietf.org/rfc/rfc4086.txt (Zugriff am 26.2.2012).

6.2.2 Daten-/Policy-Verwaltung

Die Komponente zur Daten-/Policy-Verwaltung dient dem Binden der Policies an die zu übermittelnden Datensätze. Sie verfügt über eine eingehende externe Schnittstelle für den Import der veröffentlichten Zweckdefinitionen. Intern bezieht sie die Datensätze aus dem persönlichen Datenspeicher. Der Dateneigner wählt die aktuell relevanten Datentypen sowie die dazu hinterlegten Werte aus und versieht sie mit dem zulässigen Verarbeitungszweck und der Empfänger-ID[226].

6.2.3 Verschlüsselung, Paketierung und Übermittlung

Diese Komponente stellt den zuvor erstellten Datenpaketen die korrespondierenden Schlüsselpakete zur Seite, signiert und verschlüsselt die Daten und sendet beide Pakete an die zentrale Datenverwaltung, die beim Aufbewahrer residiert.

Für die Aufgaben der Signatur und Verschlüsselung greift sie auf eine weitere interne Komponente zu, die Schlüsselverwaltung. Sie führt die für die einzelnen Datensätze individuell erstellten symmetrischen Schlüssel, das asymmetrische Schlüsselpaar des Dateneigners sowie die öffentlichen Schlüssel aller vorgesehenen Datenverarbeiter.

6.2.4 Protokollsystem

Das Protokollsystem ist auf die Systeme des Dateneigners und des Aufbewahrers verteilt. Während in der Aufbewahrungsstelle alle Existenz- und Inhaltsabfragen sowie die erfolgten Übermittlungen mit Zeitstempel in einer Datenbank protokolliert werden, verfügt der Dateneigner über eine eigene Komponente zur Abfrage und Auswertung der Protokolle, die ihn betreffen. Diese könnte auch rein Server-basiert implementiert sein, eine lokale Speicherung erscheint jedoch für die Aufbewahrung der Protokolle, beispielsweise zum Zwecke künftiger Beweisführung, sinnvoller.

6.2.5 Katalogverwaltung

Als Basis des Protokolls müssen die von allen Parteien akzeptierten Kataloge der Datentypen und Verwendungszwecke vorliegen. Dabei ist es unwesentlich, durch wen die Katalogdefinitionen erstellt wurden[227], die Aufbewahrungsinstanz eignet sich jedoch als

[226] vgl. den Vorschlag eines „Privacy Control Panel“ in [ABGG+02].

[227] Die vorangehende Diskussion zur Möglichkeit von Chosen-Plaintext-Angriffen macht jedoch eine Trennung der Verantwortlichkeiten in der Form sinnvoll, dass der Formulierer der Datentyp- und Verwendungszweck-Kataloge kein Teilnehmer am operativen Protokoll, insbesondere keine Aufbewahrungsinstanz und kein Datenverarbeiter sein sollte.

die Stelle, die den von ihr unterstützten Standard veröffentlicht und so den anderen Parteien bekannt macht.

Die Kataloge der gültigen Datentypen und Verwendungszwecke werden vom Aufbewahrer zum einen publiziert, zum anderen auch als Datei zum Import in die Applikationen von Dateneignern und -verarbeitern bereit gestellt.

6.2.6 Zentrale Datenverwaltung und Speicherung

Die zentrale Datenverwaltung residiert in der Aufbewahrungsstelle. Sie nimmt die Schlüssel- und Datenpakete über die Schnittstelle aus der Paketierungskomponente beim Dateneigner entgegen und legt sie in einem gesicherten Speicher ab.

Das Abfragesystem als weitere Komponente handhabt alle Anfragen, die von Datenverarbeitern an die Aufbewahrungsinstanz gestellt werden. Es nimmt sowohl Existenz- als auch Inhaltsabfragen entgegen, führt die entsprechenden Suchen und Vergleichsoperationen auf den gespeicherten Schlüsselpaketen durch, meldet Erfolg oder Misserfolg der Anfrage an den Verarbeiter zurück beziehungsweise reicht das identifizierte Datenpaket an diesen weiter. Das Abfragesystem verfügt über die einzige ausgehende Schnittstelle, über die personenbezogene Daten abgegeben werden.

Alle Aktivitäten des Abfragesystems werden an das Protokollsystem gemeldet, das im Protokollspeicher eine vollständige Historie aller Abfragen und Rückmeldungen vorhält. Diese Daten können vom Dateneigner abgefragt und lokal gespeichert werden. Nachträgliche Manipulationen am Protokollspeicher sollen verhindert werden[228]. Log-Anfragen von Betroffenen beantwortet die zentrale Datenbank mit einer Liste aller erfolgten Existenz- und Inhaltsabfragen, sowie der Kennung der gegebenenfalls übermittelten Datenpakete. Alle Protokolleinträge sind mit Herkunftsnachweis, Datum und Uhrzeit versehen, sowie dem Kennzeichen, ob eine Anfrage positiv oder negativ beschieden wurde.

6.2.7 Abfragewerkzeug

Applikationen des Datenverarbeiters, die Zugriff auf personenbezogene Daten des Dateneigners benötigen, stellen ihre Anfrage über den Abfrage-Client, der auf einem lokalen System des Verarbeiters liegt. Dieser prüft für jede Anfrage, ob die benötigten Daten bereits im lokalen Speicher vorliegen und reicht sie an die Applikation weiter, sofern sie schon im lokalen Speicher vorhanden sind. Werden Existenznachweise oder

[228] Hierbei sind TrueWORM-Speichertechniken einzusetzen. WORM steht für „*write once read multiple (times)*“ und ist insbesondere die Basis für digitale Archive, die Regeln zur Unveränderbarkeit ihrer Inhalte unterliegen. TrueWORM ist dabei die Variante, bei der die Unveränderbarkeit durch die physikalischen Eigenschaften des Archivierungsmediums bzw. des Schreibvorgangs hergestellt wird. Populäre Beispiele für TrueWORM sind CD-R und DVD-R/DVD+R.

Datenpakete benötigt, die noch nicht lokal abgelegt sind, verbindet sich der Abfrage-Client mit dem Abfragesystem des Aufbewahrers, spezifiziert seine Anforderungen in Form von Existenz- oder Inhaltsabfragen mit den entsprechenden Zwecknachweisen gegenüber diesem und legt die zurückerhaltenen Daten in seiner Datenbank ab. Von dort werden im Anschluss die Anfragen der Datenverarbeiter-Applikationen beantwortet.

6.2.8 Applikationsarchitektur Szenario 1

Abbildung 17 zeigt das Zusammenspiel der vorstehend erläuterten Komponenten. Dabei wird von einer bereits etablierten sicheren Netzwerk-Verbindung zwischen den beteiligten Parteien ausgegangen, die insbesondere gegenüber Man-in-the-Middle-Angriffen stabil ist.

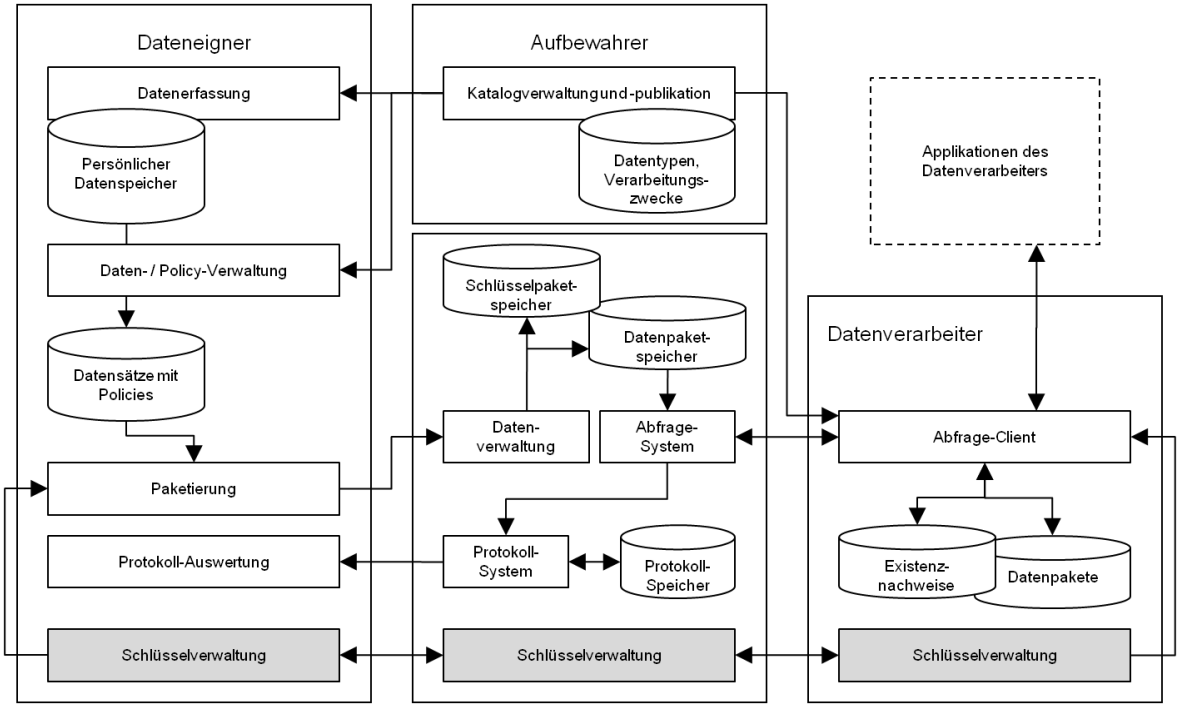


Abbildung 17: Applikationsarchitektur PDG

6.2.9 Schlüsselverwaltung

Die geschilderten Komponenten greifen an mehreren Stellen des Protokolls auf die Schlüsselverwaltung zurück. Hier liegen alle Schlüssel, die zur Durchführung notwendig sind bzw. während der Durchläufe des Protokolls anfallen und zwischengespeichert werden müssen, im verwendungsfähigen Klartext. Die Schlüsselverwaltung ist dabei für jede der beteiligten Parteien individuell ausgelegt, wie Abbildung 18 zeigt.

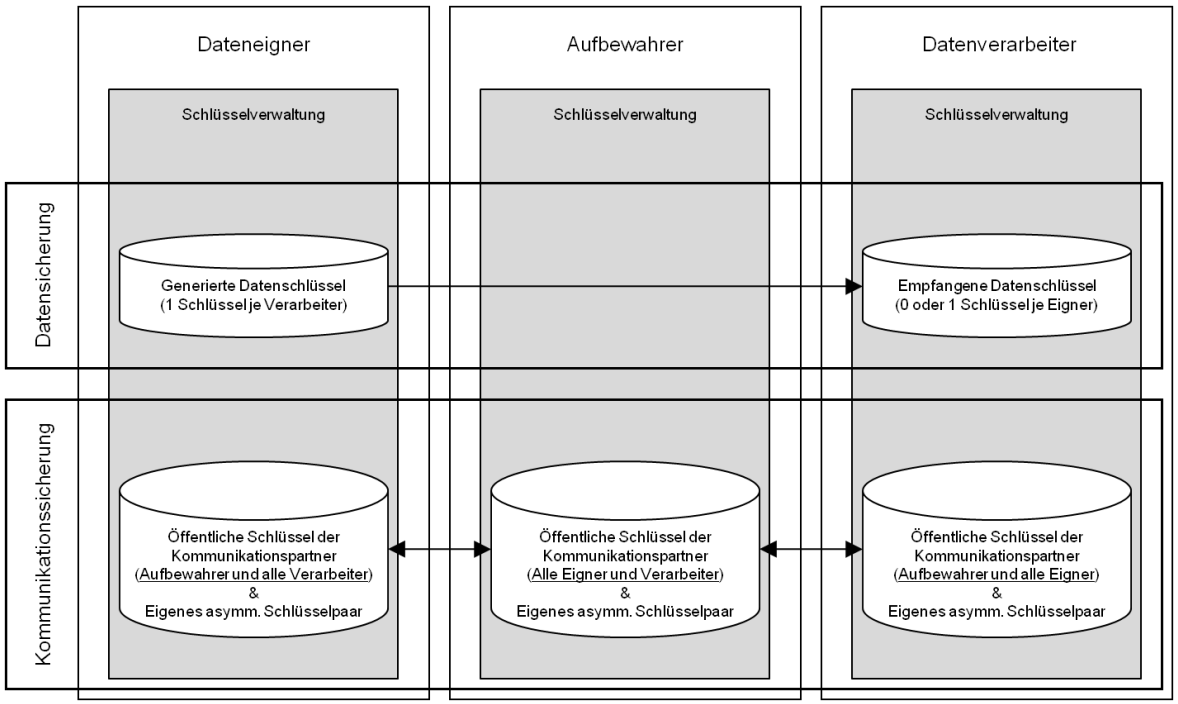


Abbildung 18: Schlüsselverwaltung PDG

Im Bereich der Kommunikationssicherung, das heißt dem grundsätzlichen Schutz aller Kommunikation, die zwischen den Beteiligten stattfindet, vor Abhören und Verfälschen, gleicht sich die Schlüsselverwaltung für alle Parteien. Jeder Teilnehmer bewahrt hier sein persönliches Schlüsselpaar für asymmetrisch verschlüsselte Kommunikation und lagert zusätzlich die öffentlichen Schlüssel seiner Kommunikationspartner. Für Dateneigner sind das die öffentlichen Schlüssel des Aufbewahrers und aller von ihm identifizierten potentiellen Verarbeiter. Der Aufbewahrer muss die öffentlichen Schlüssel aller Dateneigner und aller Verarbeiter kennen, die am System teilnehmen. Und die Verarbeiter müssen über die öffentlichen Schlüssel des Aufbewahrers und der für sie relevanten Dateneigner verfügen. Zwar sieht das Protokoll keine direkte Kommunikation vom Verarbeiter zum Dateneigner vor, die eine Verschlüsselung mit dessen öffentlichem Schlüssel erfordern würde, aber die Verarbeiter benötigen sie dennoch zur Verifizierung der digitalen Signaturen. Auf diese Weise können sie sich versichern, dass die übermittelten Inhalte vom Eigner stammen. Gleichzeitig können Eigner ihre Urheberschaft der signierten Daten gegenüber den Empfängern nicht abstreiten.

Auf der Sicherung der Kommunikationskanäle setzt die Sicherung der Daten selbst auf. Die personenbezogenen Daten der Dateneigner sind mit den symmetrischen Datenschlüsseln verschlüsselt, die zunächst nur den Eignern bekannt sind und nach erfolgreicher Schlüsselabfrage auch den jeweiligen Verarbeitern zur Verfügung stehen. Der Verarbeiter legt die empfangenen Datenschlüssel in seiner Schlüsselverwaltung ab, aus der er den Schlüssel für alle künftigen Existenz- und Inhaltsabfragen nutzt. Auch beim Aufbewahrer liegen die Datenschlüssel in verschlüsselter Form, jedoch nicht im Klartext vor. Sie sind keine von ihm anwendbaren Schlüssel und damit auch nicht in der Schlüsselverwaltung abgelegt.

6.3 Architektur für Szenario 2

Im Szenario 2 (PDG/v, mit variablen Empfängergruppen) wird neben der Aufbewahrungsstelle der von ihr unabhängige Zertifizierer etabliert. Das gegenüber Szenario 1 erweiterte Protokoll erfordert neben der beim Zertifizierer eingesetzten Software auch Anpassungen der anderen Applikationen gegenüber deren Spezifikation für Szenario 1. Abbildung 19 zeigt die Applikationsarchitektur für Szenario 2.

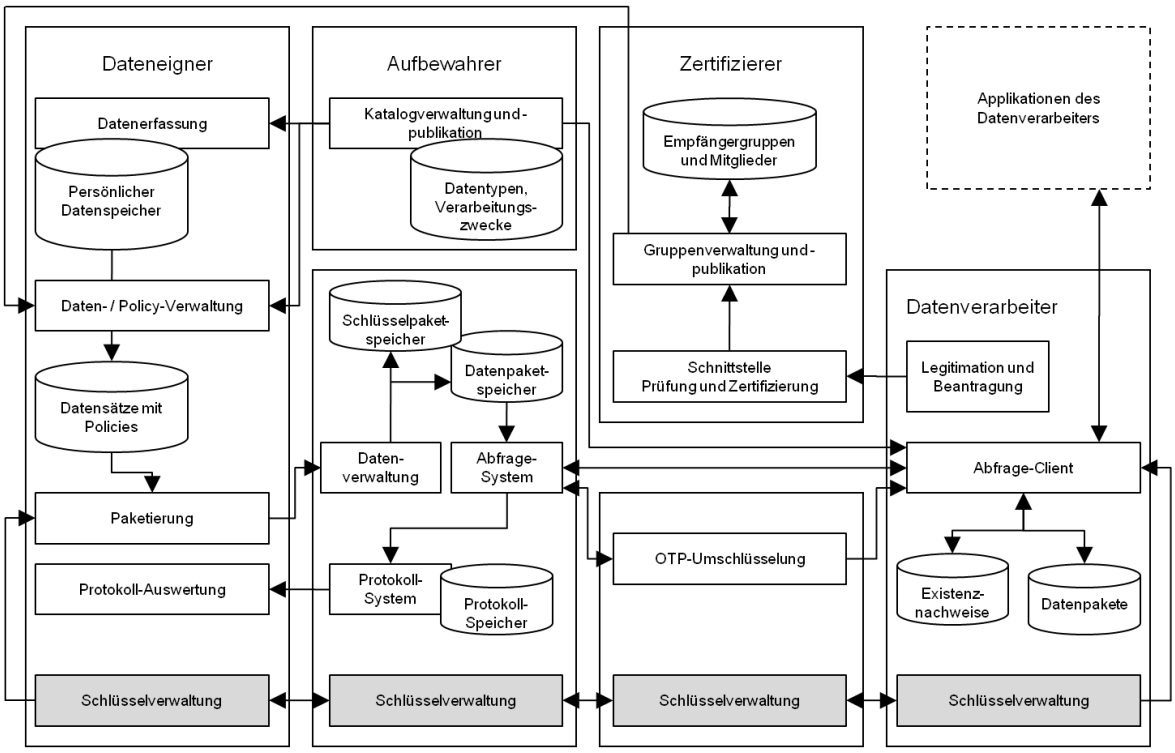


Abbildung 19: Applikationsarchitektur PDG/v

6.3.1 Anpassungen beim Dateneigner

Im System des Dateneigners sind die Applikationen zur Datenerfassung und zum Policy-Management gegenüber Szenario 1 dahingehend anzupassen, dass anstelle der direkten Verknüpfung der Schlüssel- und Datensätze an die Verarbeiter jetzt eine Bindung an Empfängergruppen stattfindet. Hierfür wird auch eine zusätzliche Schnittstelle benötigt, mittels derer die Empfängergruppendefinitionen der Zertifizierungsinstanz importiert werden. Alle Policy-Definitionen verwenden nun die Empfängergruppen anstelle der definierten Verarbeiter in Szenario 1.

Die wesentliche Änderung beim Dateneigner erfolgt in der Paketierung, die neben der Erstellung und Ablage von Schlüssel- und Datenpaketen beim Aufbewahrer nun auch die Aufgabe hat, die empfängergruppenspezifischen One-Time-Pads für den Zertifizierer zu

erzeugen und an diesen zu übermitteln. Neben einem Modul, das zuverlässige One-Time-Pads generiert[229], benötigt sie also auch eine Schnittstelle zur gesicherten Kommunikation mit der Zertifizierungsinstanz.

6.3.2 Anpassungen an der zentralen Datenverwaltung

Der Aufbewahrer verfügt über zwei Aufgaben. Zum einen ist dies die Verwaltung und Publikation der Datentyp- und Verarbeitungszweck-Definitionen, die unverändert aus Szenario 1 übernommen wird.

Zum anderen ist es die Verwahrung und Weitergabe der Datenpakete, die von den Eignern zusammengestellt wurden. Die Komponente für die Datenverwaltung erfordert keine besondere Anpassung für Szenario 2, die Ablagestrukturen behalten ihre Gültigkeit. Wie bereits in Szenario 1 sind die Schlüssel- und Datenpakete für den Aufbewahrer ohne inhaltliche Bedeutung. Er kann sie nicht entschlüsseln und ohne das Vorliegen konkreter Anfragen weder dem Eigner noch künftigen Verarbeitern zuordnen. Dass die in den Datenpaketen enthaltenen Policies und die Schlüsselpakete in Szenario 2 auf Empfängergruppen anstelle individueller Verarbeiter abgestellt sind, ist für die Aufbewahrung ohne Bedeutung.

Das ändert sich, sobald eine Schlüsselabfrage zu beantworten ist. Jetzt muss das Abfragesystem zunächst die Autorisierung des Anfragenden als zertifiziertes Gruppenmitglied überprüfen. Dazu fragt es die aktuellen Mitgliederlisten der Empfängergruppen am Publikationsort des Zertifizierers ab. Nach erfolgter Autorisierung lässt es die empfängerspezifischen One-Time-Pads generieren und führt die XOR-Verschlüsselung des Schlüsselpakets mit eben diesem One-Time-Pad durch. Es folgt die Übermittlung, einerseits des zweifach verschlüsselten Schlüsselpakets an den Zertifizierer, und andererseits des empfängerspezifischen One-Time-Pads an den jeweiligen Empfänger.

6.3.3 Zertifizierung und Gruppenverwaltung

Die eigentliche Zertifizierung umfasst zwei Komponenten, die schließlich in einer veröffentlichten Datenbank münden. In dieser Datenbank werden alle definierten Empfängergruppen und ihre aktuell gültigen Mitglieder geführt.

Die Aufnahme als Mitglied einer Empfängergruppe erfordert eine entsprechende Legitimation, die der Kandidat gemeinsam mit einem „Aufnahmeantrag“ bei der Zertifizierungsinstanz einreicht. Je nach Ausgestaltung der Aufnahmekriterien kann die Schnittstelle auf Seiten des Zertifizierers mehr oder weniger komplex ausfallen. Sie kann möglicherweise manuelle Kontrollen, Prüfung gegen externe Datenbanken, Wartefristen

[229] Das bedeutet, das Modul muss Bitfolgen erzeugen können, die hinreichend zufällig sind.

oder ähnliches beinhalten.[230] An deren Ende steht in jedem Fall entweder die Information des Kandidaten über eine Ablehnung oder die Annahme des Antrags und damit einhergehend die Meldung des neuen Gruppenmitglieds an die Gruppenverwaltung. Diese Komponente ist dafür zuständig, zu jedem Zeitpunkt das aktuelle Verzeichnis aller Gruppenmitgliedschaften bereit zu halten und zeitnah zu veröffentlichen, so dass es für Applikationen und Dienste der Dateneigner und insbesondere der Aufbewahrungsinstanz nutzbar ist.

6.3.4 Umschlüsselung

Getrennt von der Empfängergruppenverwaltung liegt die Umschlüsselung der Schlüsselpakete beim Zertifizierer. Ihre Funktion besteht darin, die zweifach verschlüsselten Schlüsselpakete, die sie vom Aufbewahrer erhält, mit dem passenden empfängergruppenspezifischen One-Time-Pad zu verknüpfen, das sie von dem Dateneigner geschickt bekommen und in der Schlüsselverwaltung abgelegt hat. Dadurch löst sie die ursprüngliche Verschlüsselung des Dateneigners auf, das Schlüsselpaket bleibt dennoch durch das vom Aufbewahrer erzeugte One-Time-Pad chiffriert und damit vor dem Zertifizierer verborgen.

6.3.5 Abfragewerkzeug

Die Applikationen des Datenverarbeiters, die Zugriff auf personenbezogene Daten des Dateneigners benötigen, stellen ihre Anfrage über den Abfrage-Client. Wie in Szenario 1 korrespondiert der Abfrage-Client mit dem Abfragesystem des Aufbewahrers für Schlüssel-, Existenz- und Inhaltsabfragen. Die Schlüsselabfrage wird jedoch über den Umweg des Zertifizierers zum Zweck der geschilderten One-Time-Pad-Umschlüsselung durchgeführt. Entsprechend verfügt der Abfrage-Client der Datenverarbeiter in Szenario 2 auch über die zusätzliche Fähigkeit, die XOR-Verknüpfungen mit dem empfängerspezifischen One-Time-Pad durchzuführen, das durch den Aufbewahrer erstellt und übermittelt wurde.

6.3.6 Erweiterte Schlüsselverwaltung

Abbildung 20 zeigt die erweiterte Schlüsselverwaltung. Neben den aus Szenario 1 bekannten Ablagen für die zur Kommunikationssicherung erforderlichen Schlüssel zur

[230] Die Zertifizierung als Gruppenmitglied ist in diesem Sinne ähnlich der Zertifizierung von öffentlichen Schlüsseln. Hier erfolgt die Zertifizierung meist durch Vorzeigen von Ausweispapieren, etwa bei der Krypto-Kampagne des Heise-Verlags (http://www.heise.de/security/dienste/Krypto-Kampagne-2111.html, Zugriff: 21.05.2012), im PostIdent-Verfahren oder auch durch notarielle Beurkundung. Die Zertifizierung als Mitglied einer Verarbeitergruppe wird über die Identifikation hinaus weitere Nachweise beispielsweise zu Qualifikation, Position oder Unternehmenssitz eines Verarbeiters erfordern.

asymmetrischen Kryptographie und den Datenschlüsseln, wird eine dritte Schicht etabliert. Sie dient der Sicherung des Schlüsselaustauschs. Dateneigner und Zertifizierer speichern hier die empfängergruppenspezifischen One-Time-Pads, während der Aufbewahrer und Verarbeiter in dieser Schicht die empfängerspezifischen One-Time-Pads lagern.

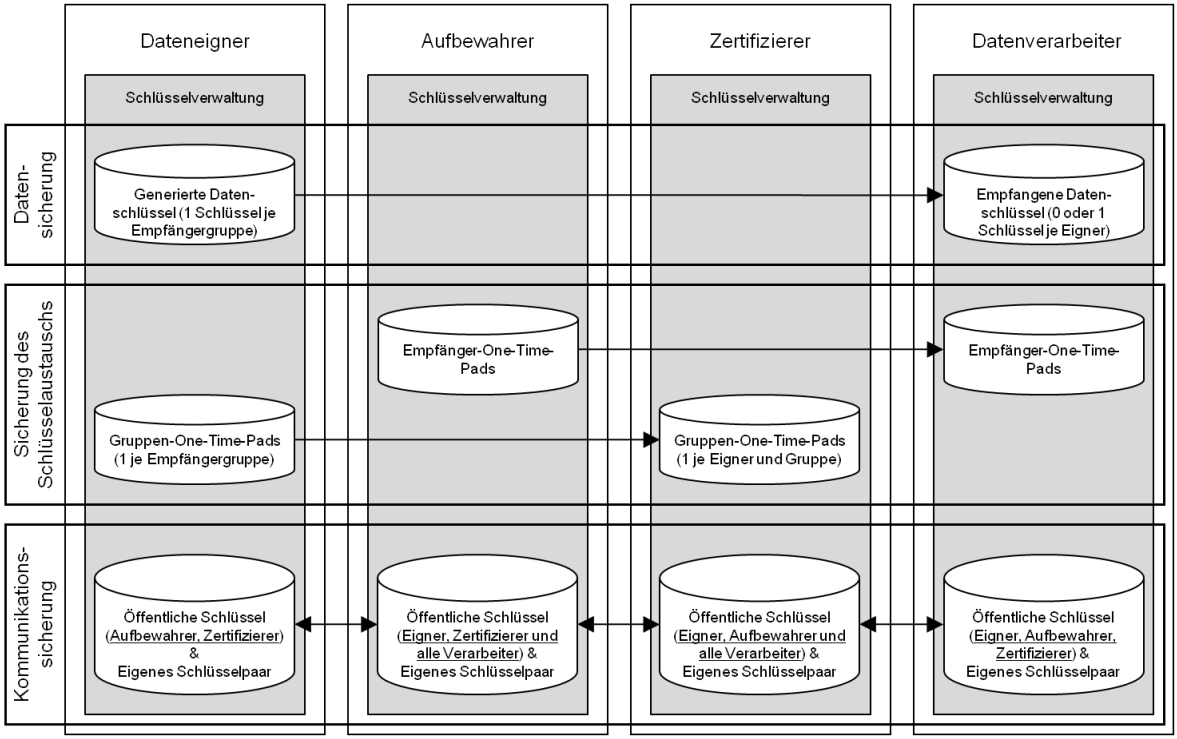


Abbildung 20: Schlüsselverwaltung PDG/v

Gemäß den Prämissen des Szenarios 2 speichern die Empfänger in der Schicht zur Kommunikationssicherung natürlich nicht die öffentlichen Schlüssel der Verarbeiter (die sie nicht kennen), sondern den öffentlichen Schlüssel des Zertifizierers und analog zu Szenario 1 den des Aufbewahrers. Der Zertifizierer kommuniziert im Verlauf des Protokolls mit allen anderen Parteien und benötigt daher die öffentlichen Schlüssel von allen. Dasselbe gilt für den Aufbewahrer, da er nach erfolgreicher Schlüsselabfrage direkt mit den Empfängern kommuniziert[231].

Die Schicht zur Datensicherung speichert unverändert zu Szenario 1 die Datenschlüssel, die der Eigner zur symmetrischen Verschlüsselung seiner Daten verwendet hat und die nach erfolgreicher Schlüsselabfrage dem Verarbeiter vorliegen, um für dessen weitere

[231] Alternativ könnte alle Kommunikation zwischen Aufbewahrer und Verarbeitern weiterhin über die Zertifizierungsinstanz laufen. Dies wäre jedoch nur dann von Wert, wenn der Zertifizierer keine Veröffentlichung der Mitgliedslisten der Empfängergruppen vornehmen will oder darf, oder wenn die Mitgliedslisten so dynamisch wären, dass eine Veröffentlichung in Echtzeit nicht machbar wäre. In diesem Fall wäre jede Anfrage und auch jede Antwort vor ihrer Weitergabe an den Aufbewahrer beziehungsweise den Verarbeiter beim Zertifizierer während der Durchleitung erneut daraufhin zu prüfen, ob die Gruppenmitgliedschaft des Anfragenden noch gültig ist.

Existenz- und Inhaltsabfragen zu dienen. Wiederum werden diese Schlüssel im Verlauf des Protokolls auch bei Aufbewahrer und Zertifizierer gespeichert, allerdings nicht als verwendbare Schlüssel, sondern nur in Form von umzuschlüsselnden und weiterzuleitenden Paketen. Daher sieht die Schlüsselverwaltung dieser beiden Parteien keine Ablage für die Datenschlüssel vor.

Die One-Time-Pads liegen wie beschrieben in den Schlüsselverwaltungen der Parteien, wobei Aufbewahrer und Datenverarbeiter das ihnen bekannte empfängerspezifische One-Time-Pad vernichten können, sobald die Schlüsselabfrage des Verarbeiters erfolgreich stattgefunden hat. Der Zertifizierer jedoch muss das empfängergruppenspezifische One-Time-Pad aufbewahren, solange noch die Möglichkeit besteht, dass weitere Mitglieder zu der betreffenden Empfängergruppe hinzustoßen können.

6.4 Implementierung an der Universität Hamburg

Im Rahmen eines Kurses am Arbeitsbereich Sicherheit in Verteilten Systemen an der Universität Hamburg implementierten Christian Baumann, Ahmed Hodjov, Stephan Hoepfner, Roman Jerger, Dennis Keitzel, Hannes Kuhlmann und Stephan Lauterbach im Sommersemester 2011 und Wintersemester 2011/12 das Verfahren im vollständigen Szenario PDG/v. Dieser Abschnitt fasst die wichtigsten dabei getroffenen Implementierungsentscheidungen zusammen und zeigt Abbildungen aus den Applikationen für Dateneigner und Verarbeiter.

Die Sicherung der Kommunikation zwischen den beteiligten Parteien erfolgt per Secure Socket Layer (SSL). Für die symmetrischen Verschlüsselungen wurde AES[232] mit 256 Bit Schlüssellänge gewählt. Die benötigte Public Key Infrastruktur basiert auf X.509 Zertifikaten mit Certificate Revocation List[233]. Für die asymmetrische Verschlüsselung findet der RSA-Algorithmus Anwendung. Um Nachrichten zu signieren, wird RSA mit dem Hashverfahren SHA-512[234] kombiniert. Das Generieren der One-Time-Pads erfolgt in dieser prototypischen Implementierung in Form von Pseudozufallszahlen aus dem Java-Zufallszahlengenerator.

Der Dateneigner installiert auf seinem Rechner eine Java[235]-Applikation, die ihm die Datenerfassung, das Erfassen und Binden der Policies und die Übermittlung an den Aufbewahrer ermöglicht. Abbildung 21 zeigt die Benutzeroberfläche, wie sie sich dem Dateneigner zum Einstieg präsentiert. Das hier exemplarisch verwendete Anwendungsszenario entstammt dem medizinischen Bereich. Im Abschnitt „Overview“ links oben lassen sich die bereits angelegten Datensätze überblicken, bei Anwahl eines Datensatzes werden dessen Details und die Protokolle bereits stattgefundener Existenz- und Inhaltsabfragen sichtbar. Oben rechts zeigt das Fenster für

[232] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (Zugriff am 05.05.2012).
[233] http://tools.ietf.org/html/rfc5280 (Zugriff am 05.05.2012).
[234] http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf (Zugriff am 05.05.2012).
[235] http://www.java.com/de/ (Zugriff am 05.05.2012).

jeden Datensatz die Policy, bestehend aus Verarbeitungszweck („Purpose“), Datentyp („Type“) und zulässiger Verarbeitergruppe („Group“). Die Oberfläche bietet Schaltflächen zur Neuanlage eines Datensatzes, zum Ändern und Löschen von Datensätzen sowie zur Aktualisierung der Abfrage-Protokolle. Das Feld „Data“ unten rechts enthält die zu schützenden personenbezogenen Daten. Wie im Protokoll zu PDG vorgesehen, kann der Datenbereich in dieser Form beliebigen Text aufnehmen. In künftigen Versionen könnte dies um die Möglichkeit der Hinterlegung von strukturierten Daten wie ausgefüllten Formularen oder Binärdateien wie Röntgenbildern erweitert werden.

Möchte ein Dateneigner einen neuen Datensatz ablegen, wählt er im Übersichtsbild das Plus-Symbol und nutzt das in Abbildung 22 gezeigte Fenster zur Datenerfassung. Hier kann er aus den veröffentlichten Definitionen zu Datentypen und Verarbeitungszwecken die für seine Policy geeigneten auswählen, die Empfängergruppe festlegen und die Nutzdaten eintragen.

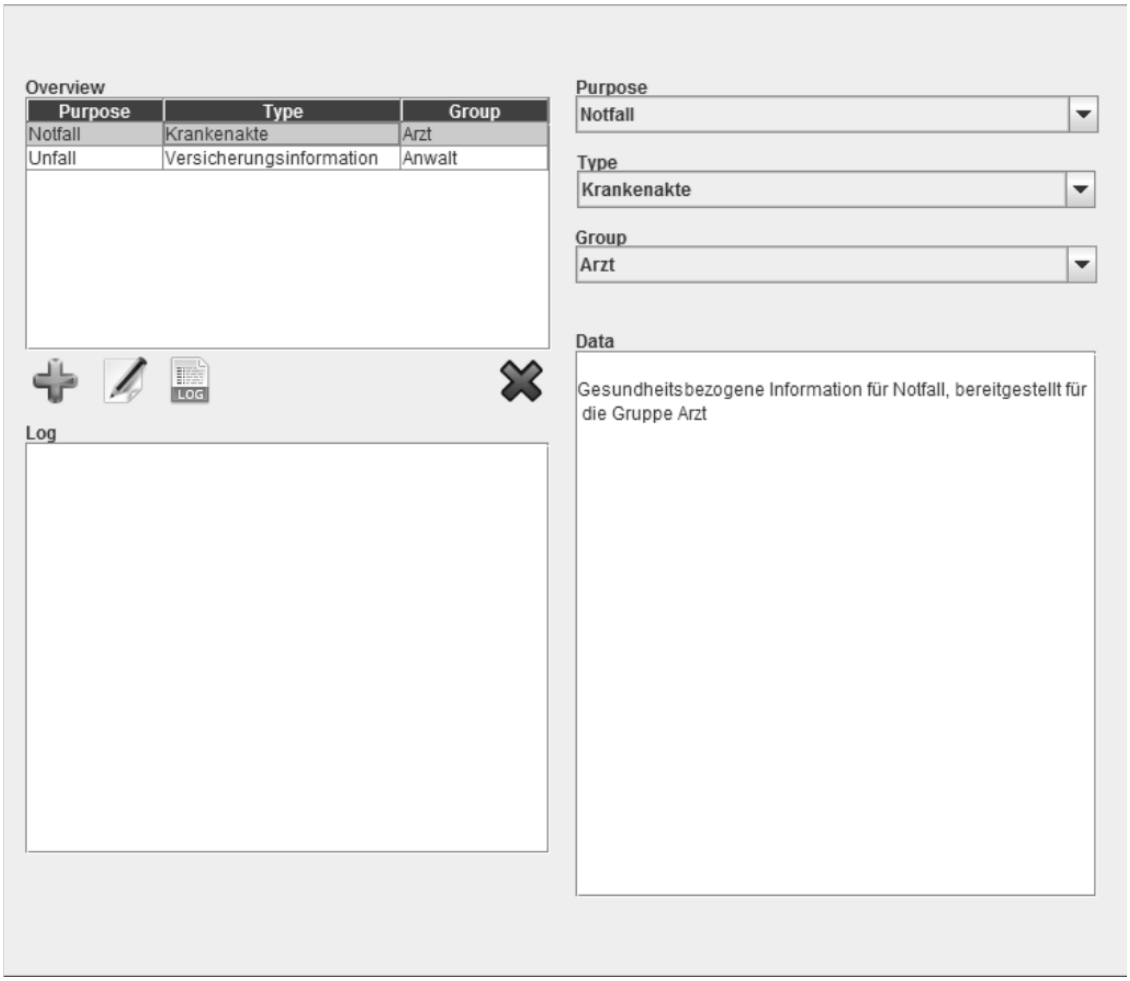


Abbildung 21: Benutzeroberfläche für den Dateneigner[236]

[236] Alle Abbildungen in Kapitel 6.4 sind der Dokumentation zum Masterprojekt entnommen [BHHJ+12].

Abbildung 22: Datenerfassung durch den Dateneigner

Analog verhält es sich bei der Änderung bestehender Datensätze, nur dass hier die Policy-Felder für Datentyp und Verarbeitungszweck nicht änderbar sind. Wären sie zu ändern, käme das der Neuaufnahme eines Datensatzes mit einer anderen Policy gleich. Es können aber durchaus neue Empfängergruppen hinzugefügt werden.

Durch Betätigen der Schaltfläche für die Logabfrage werden die Protokolle aller bereits erfolgten Existenz- und Inhaltsabfragen zum gewählten Datensatz geladen und angezeigt. Die Anzeige umfasst die Information, um welche Art der Abfrage es sich gehandelt hat, von wem die Anfrage gestellt wurde und zu welchem Zeitpunkt sie erfolgte. Abbildung 23 zeigt ein beispielhaftes Protokoll.

Zuletzt bietet die Benutzeroberfläche des Dateneigners auch die Option, Datensätze zu löschen. Dazu werden sie im Überblicksfeld markiert und durch Betätigen der Löschen-

Schaltfläche aus der Datenablage entfernt. War ein Datensatz für die gleiche Kombination von Verwendungszweck und Datentyp an mehrere Empfängergruppen freigegeben, wird er in der vorliegenden Implementierung für alle Gruppen zugleich gelöscht.

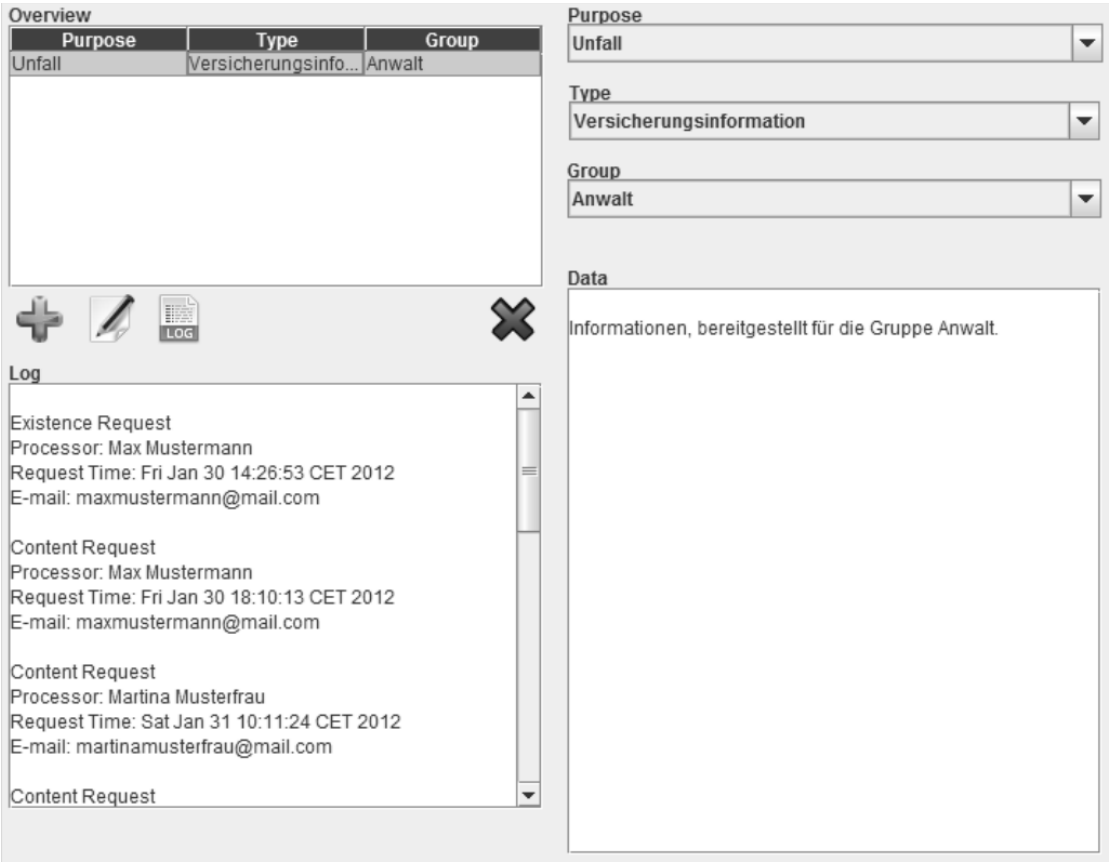


Abbildung 23: Anzeige der Zugriffsprotokolle

Die mit der Implementierung befasste Projektgruppe schlägt zur Weiterentwicklung vor, die Java-Applikation des Dateneigners künftig als Webanwendung zu gestalten, so dass der Eigner von überall auf seine Daten zugreifen und falls nötig Änderungen und Ergänzungen vornehmen kann, ohne auf eine bestimmte Infrastruktur angewiesen zu sein.

Die lokale Datensammlung des Eigners wird von der Applikation in einer kompakten, dateibasierten HSQL-Datenbank[237] gehalten. Die Datenbanken von Aufbewahrer und Zertifizierer sind dem gegenüber unter dem Open Source Datenbankmanagementsystem PostgreSQL[238] implementiert, bei dem Skalierbarkeit und Performanz im Vordergrund stehen. MyBatis[239] dient als Datenbank-Framework zur Trennung der SQL-Kommandos vom Java-Applikationscode. Der Datenaustausch zwischen den Anwendungen nutzt das

[237] http://www.hsqldb.org/ (Zugriff am 05.05.2012).
[238] http://www.postgresql.org/ (Zugriff am 05.05.2012).
[239] http://www.mybatis.org/ (Zugriff am 05.05.2012).

Datenformat JavaScript Object Notification (JSON)[240], das sich durch Kompaktheit und für Mensch und Maschine einfach lesbare Textform auszeichnet.

Als Programmiersprache für die lokalen Module bei Aufbewahrer und Zertifizierer wurde ebenfalls Java gewählt. Der Java Code wird unter einem Tomcat-Applikationsserver[241] an einem Apache-Webserver[242] ausgeführt.

Die Software des Datenverarbeiters ist als Webanwendung implementiert und kann in beliebigen JavaScript-fähigen Umgebungen genutzt werden. Dazu muss sich das Gerät der Wahl einmalig mit einem Webserver verbinden und von diesem die Anwendung laden, die Elemente des JavaScript Frameworks ExtJS[243] nutzt. Danach steht ihm für seine Abfragen die Benutzeroberfläche zur Verfügung, wie sie in Abbildung 24 zu sehen ist[244]. Sie bietet auf der linken Seite einen Navigationsbereich, der zur Suche nach den gewünschten Datensätzen, also dem Formulieren und Senden der Vergleichs-Policies dient. Die rechte Seite ist für die Anzeige der Abfrageergebnisse reserviert.

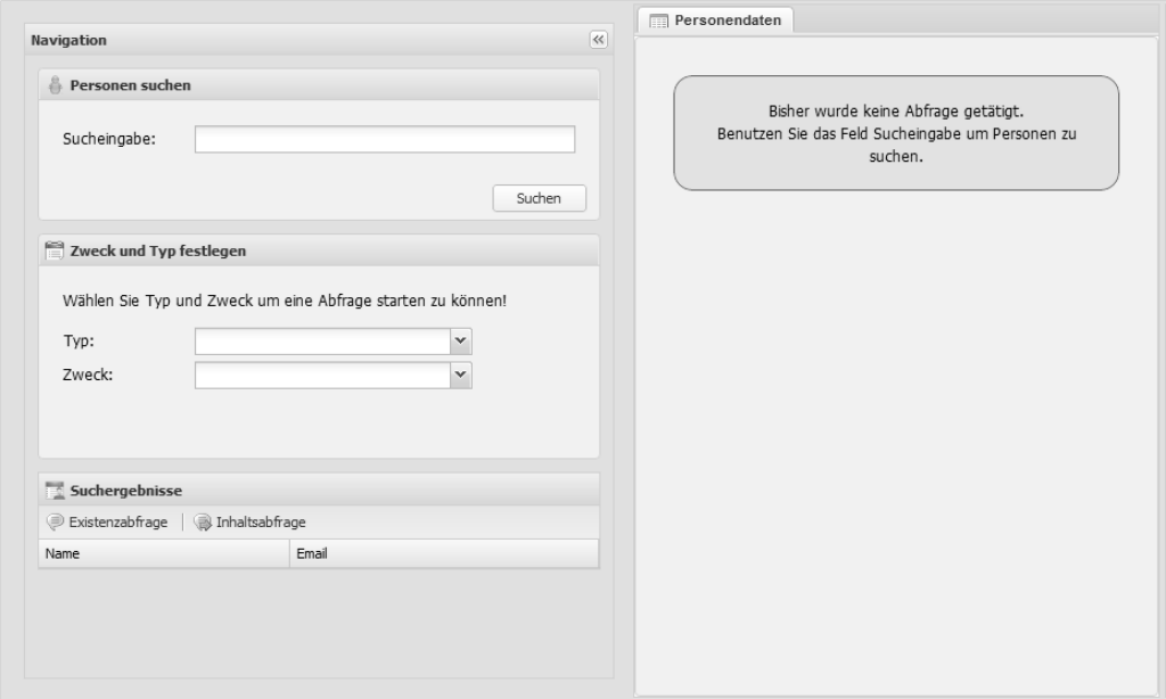


Abbildung 24: Startoberfläche für den Verarbeiter

[240] http://www.json.org/ (Zugriff am 05.05.2012).
[241] http://tomcat.apache.org/ (Zugriff am 05.05.2012).
[242] http://www.apache.org/ (Zugriff am 05.05.2012).
[243] http://docs.sencha.com/ext-js/4-1/ (Zugriff am 05.05.2012).
[244] Die Flexibilität der Nutzung des Verfahrens auf verschiedenen Endgeräten des Verarbeiters erfordert jedoch im Gegensatz zu einer stationären Schlüsselverwaltung, dass jeder Existenz- und Inhaltsabfrage eine neuerliche Schlüsselabfrage vorangeht.

Im Bereich „Personen suchen“ wählt der Verarbeiter zunächst den Dateneigner aus, für den er eine Abfrage formulieren möchte. Diese spezifiziert er im Bereich „Zweck und Typ festlegen“ durch die Auswahl der jeweiligen Elemente aus den veröffentlichten Vokabularen für Datentypen und Verarbeitungszwecke, wie es Abbildung 25 zeigt.

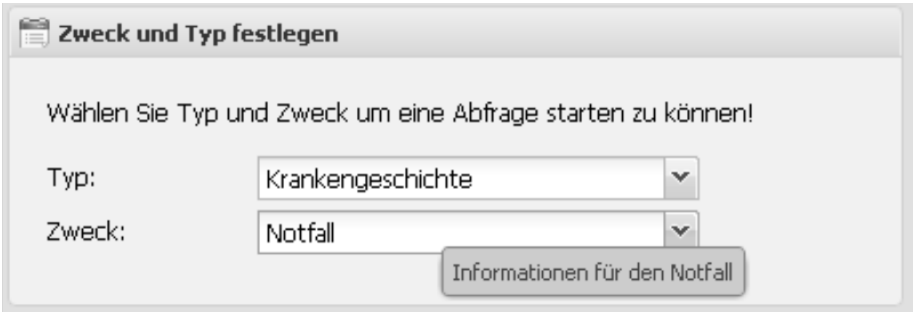


Abbildung 25: Auswahl der Abfragepolicy

Damit ist die Vergleichs-Policy erstellt und im nächsten Schritt (Abbildung 26) erfolgt die Auswahl, ob nur die Existenz eines zur Policy beim Aufbewahrer hinterlegten Datensatzes nachgewiesen oder der Datensatz selbst abgefragt werden soll.

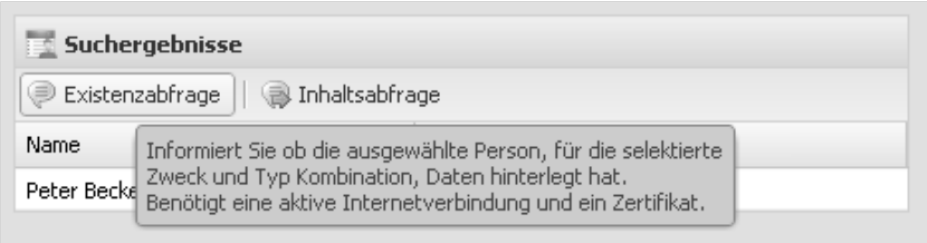


Abbildung 26: Auswahl des Abfragetyps

Das Ergebnis einer beispielhaften Existenzabfrage mit negativem Ergebnis zeigt Abbildung 27, während Abbildung 28 die Rückmeldung einer erfolgreichen Inhaltsabfrage demonstriert. Die gezeigten Abfrageergebnisse sind hier in der Form ausgegeben, wie sie vom Verarbeiter auf dessen Endgerät gelesen werden können. Analog kann aber ebenso das Weiterreichen der Ergebnisse an weiterverarbeitende Geschäftsapplikationen des Verarbeiters, wie ein Personalverwaltungssystem oder eine Software zum Patientenmanagement erfolgen.

Abbildung 27: Ergebnis einer misslungenen Existenzabfrage

Abbildung 28: Ergebnis einer erfolgreichen Inhaltsabfrage

7 Analyse des Verfahrens

Die folgenden Abschnitte skizzieren Angriffe auf das Protokoll, die durch die Beteiligten oder auch durch außenstehende Parteien durchgeführt werden könnten. Dies umfasst nicht nur böswillige Attacken, sondern auch möglicherweise ungewolltes Fehlverhalten.

Die Angriffe werden im Folgenden nach ihrer Quelle gegliedert. Begonnen wird mit den Dateneignern, gefolgt vom Aufbewahrer, speziell für Szenario 2 der Zertifizierungsinstanz und schließlich den Verarbeitern. Hierbei gilt zunächst die Annahme, dass alle Kommunikation zwischen den Beteiligten sicher ist. Diese Annahme wird dann in Kapitel 7.5 aufgehoben, das Angriffe durch Außenstehende analysiert. Hierunter fallen auch Angriffe durch Beteiligte des Protokolls, sofern sie sich gegen die zugrundeliegende Kommunikation richten.

Für jedes Angriffsszenario werden die Attacke selbst, die mögliche Motivation und der Ausführungsmodus beschrieben. Dem folgt die Diskussion der Verteidigungsmaßnahmen. Jedem Angriff werden dabei zusätzlich die Datenschutzziele zugeordnet, gegen die er vornehmlich gerichtet ist. Die Schutzziele sind[245]:

- Integrität: Der Schutz der personenbezogenen Daten bzw. der aus der Protokollanwendung angestrebten Informationen gegen Verfälschung oder Verlust.
- Verbindlichkeit: Die Nachweisbarkeit der durchgeführten Aktivitäten der Beteiligten, der Urheberschaft der Daten und des Datenabrufs.
- Vertraulichkeit: Der Schutz der übermittelten Daten und anderer aus dem System zu gewinnender Erkenntnisse vor der Kenntnisnahme oder Nutzung durch nicht legitimierte Parteien; Zugleich auch Schutz der Zweckbindung für die Nutzung der Daten.
- Verfügbarkeit: Der Schutz des Systems und des Protokolls gegen Eingriffe, die verhindern, dass benötigte Daten und Dienste zum benötigten Zeitpunkt an der gewünschten Stelle zur Verfügung stehen.

[245] vgl. [GoKl03]

7.1 Angriffe durch Dateneigner

7.1.1 Leugnen der Datenhinterlegung und behauptete Datenmanipulation

Schutzziel

Verbindlichkeit

Motivation und Angriffsszenario

Ein Verarbeiter stellt nach erfolgter Inhaltsabfrage fest, dass die Daten auf nicht-konformes Verhalten des Eigners hinweisen und leitet rechtliche Schritte ein (z.B. bei Hinweisen auf Steuerhinterziehung oder falschen Angaben im Einstellungsgespräch). Der Eigner will diese Schritte dadurch verhindern, dass er entweder leugnet, die Daten hinterlegt zu haben, oder behauptet, ein anderer hätte im Verlauf des Protokolls die Daten zu seinem Nachteil verändert.

Verteidigung

Der Verarbeiter kann zweifelsfrei nachweisen, dass die Daten vom Eigner stammen: Der Datenschlüssel, den er beim Aufbewahrer abgerufen hat und den er zur Anforderung wie auch zur Entschlüsselung des Datenpakets verwendet hat, wurde vom Eigner generiert, was durch dessen Signatur am Datenschlüssel nachgewiesen ist. Auch das Datenpaket selbst weist die Signatur des Eigners auf, nachdem es mit dem Datenschlüssel dechiffriert wurde. Keine andere Partei kann die Signaturen hergestellt haben, sofern der Urheber seinen privaten Schlüssel geheim gehalten hat. Hat er dies nicht getan, steht der Nachweis dieser Behauptung in seiner Verantwortung.

7.1.2 Widersprüchliche Datenablage für verschiedene Verarbeiter

Schutzziel

Integrität

Motivation und Angriffsszenario

Der Eigner will sich gegenüber verschiedenen Verarbeitern unterschiedlich darstellen, etwa gegenüber Finanzbehörden seine Einkommensverhältnisse schlechter darstellen als gegenüber Kreditgebern.

Verteidigung

Diese Möglichkeit kann bei einem Verfahren, das auf Selbstauskunft beruht, nicht vollständig verhindert werden – ebenso wenig wie das grundsätzliche Lügen des Eigners bezüglich seiner Daten. Sie könnte nur dadurch begrenzt werden, dass eine weitere

vertrauenswürdige Instanz in das Protokoll eingeführt wird, die eine vorgelagerte Echtheitsprüfung der Daten aller Eigner vornimmt und dann deren Rolle bezüglich der Ablage beim Aufbewahrer übernimmt.

Dass der Eigner für verschiedene Verarbeiter bzw. Empfängergruppen zu einem bestimmten Datentyp verschiedene Daten hinterlegt, wird zumindest durch die vorgesehene Implementierung des Protokolls erschwert: Der persönliche Datenspeicher beim Eigner sieht für die personenbezogenen Daten nur die Felder „Eigner“, „Datentyp“ und „Daten“ vor. Die Zuordnung dieser Datensätze zu berechtigten Verarbeitern bzw. Empfängergruppen erfolgt erst danach in der Komponente zur Policy-Verwaltung. Hier kann der Eigner nur entscheiden, ob ein Datentyp einem Verarbeiter oder einer Empfängergruppe zu einem bestimmten Zweck zur Verfügung stehen soll oder nicht.

Wollte er im Rahmen des geschilderten Betrugsversuchs eine verarbeiterspezifisch unterschiedliche Variante des Datensatzes hinterlegen, müsste er dazu die Verarbeitungslogik und die Datenbankstruktur seiner Applikation manipulieren. Dies wiederum könnte durch eine manipulationssichere Implementierung der Applikationen oder Integritätsnachweise der Software gegenüber dem Aufbewahrer unterbunden werden. Diese zusätzlichen Sicherungsmechanismen sind eine über das vorliegende Werk hinausgehende Weiterführung der zuvor geschilderten Implementierung und Applikationsarchitektur.

7.1.3 Löschen von Datensätzen nach Existenzanfrage

Schutzziel

Verfügbarkeit

Motivation und Angriffsszenario

Der Eigner will formalen Bedingungen genügen, die erfordern, dass er bestimmte Datensätze abgelegt hat (z.B. Ablage bestimmter Daten als Einstellungsvoraussetzung), aber im Streitfall die Daten doch nicht herausgeben. Dazu wartet er ab, bis der potentielle Datenverarbeiter eine Existenzanfrage auf den betreffenden Datensatz durchgeführt hat und diese vom Aufbewahrer positiv beschieden wurde[246]. Dann sendet er dem Aufbewahrer eine Löschanforderung für den Datensatz, so dass der Verarbeiter später keine erfolgreiche Inhaltsabfrage für diese Daten absetzen kann.

Verteidigung

Es ist das Recht des Eigners, Datensätze zu entfernen. Damit dies jedoch nicht zu Täuschungen der Verarbeiter führt, überprüft der Aufbewahrer laut Protokoll alle historischen Existenzanfragen aus seinem Protokoll erneut, wenn eine Löschanforderung

[246] Um den richtigen Zeitpunkt zu ermitteln, muss er nur die Protokolle überwachen, die der Aufbewahrer von allen Anfragen anfertigt und die den betroffenen Eignern zur Prüfung zur Verfügung stehen.

durch einen Dateneigner initiiert wurde. Dazu vergleicht er die bei der Löschung als Identifikator übermittelten Policies mit den im Protokoll als bereits abgefragt gekennzeichneten Policies und informiert bei Übereinstimmung den jeweiligen Verarbeiter. Somit erfährt der Verarbeiter unmittelbar nach der Löschung davon, dass seine bisher positiv beschiedene Existenzanfrage keine Gültigkeit mehr besitzt.

7.1.4 Veranlassen der Löschung oder Änderung der Datensätze anderer Eigner

Schutzziel

Integrität

Motivation und Angriffsszenario

Ein Individuum hat sich als Dateneigner für das Protokoll registriert, um das System „von innen“ zu kompromittieren oder anderen Eignern bewusst Schaden zuzufügen. Er nutzt seine Stellung gegenüber dem Aufbewahrer, Änderungen und Löschungen von Datenpaketen veranlassen zu können. Dabei macht er sich den Umstand zu Nutze, dass der Aufbewahrer an den Datenpaketen nicht erkennen kann, zu welchem Eigner diese eigentlich gehören[247].

Verteidigung

Der Angriff scheitert, denn ein Eigner kann – ebenso wie andere Parteien – die beim Aufbewahrer gespeicherten Datensätze anderer Eigner nicht korrekt adressieren. Das ist darin begründet, dass nur der rechtmäßige Eigner die Zufallszahlen kennt, die er gemeinsam mit seinen Datenpaketen abgelegt hat. Ein Angreifer könnte nur auf sein Glück setzen und versuchen, gültige Kombinationen von Policy und passender Zufallszahl (bzw. deren Hashwert) zu raten, was bei ausreichend langen Zufallszahlen jedoch hinreichend unwahrscheinlich wird.

Auch ist es dem Angreifer nicht möglich, bereits früher durch den echten Dateneigner erfolgte Änderungsaufträge erneut in das System einzuspielen (Replay-Attacke). Dafür sorgt die Tatsache, dass Dateneigner und Aufbewahrer bei jeder Änderung die ursprünglich hinterlegte Zufallszahl durch den für die Änderungsanforderung erzeugten Hashwert ersetzen. Die nächste Anfrage müsste also den erneuten Hashwert dieses Hashwertes tragen, und nicht mehr den Hash der initial hinterlegten Zufallszahl.

[247] Das setzt voraus, dass die Dateneigner die beschriebenen Möglichkeiten nutzen, ihre Datenpakete anonym an die Aufbewahrungsinstanz zu übermitteln. Haben dem entgegen alle Eigner beschlossen, sich als Absender der Pakete zu erkennen zu geben, und würde der Aufbewahrer diese Information mitschreiben, wäre dieses Angriffsszenario hinfällig.

7.1.5 Abruf der Protokolle von anderen Eignern

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Ein Beteiligter ist neben seiner Eigenschaft als Dateneigner auch daran interessiert, Profit aus Daten über andere Eigner zu generieren. Er nutzt seine Stellung gegenüber dem Aufbewahrer, Protokollabfragen durchführen zu können. Der Aufbewahrer kann an den abgefragten Protokollsätzen nicht erkennen, für welchen Eigner diese vorgesehen sind.

Verteidigung

Ein Dateneigner übermittelt zur Abfrage seiner Protokolle die verschlüsselte Policy, für die er die Protokolle einsehen möchte. Diese vergleicht der Aufbewahrer mit seinen Log-Einträgen und liefert die Einträge zurück, bei denen die Policy-Chiffrate übereinstimmen. Außer dem Dateneigner kennt niemand den Inhalt der Policy[248], d.h. auch nicht die Information, welcher Eigner die Übermittlungen an welchen Verarbeiter einsehen möchte. Schließlich ist die Vergleichspolicy mit dem Datenschlüssel verschlüsselt, den nur Eigner und Verarbeiter kennen. Ein fremder Angreifer könnte allerdings abwarten, bis der von ihm fokussierte Dateneigner eine eigene Logabfrage stellt und dann über einen Replay-Angriff mit derselben Nachricht die Logergebnisse des anderen für sich selbst zurück erhalten. Das wird jedoch leicht dadurch verhindert, dass in das Protokoll die grundsätzliche Verschlüsselung aller Kommunikation mit den jeweiligen öffentlichen Schlüsseln der jeweiligen Empfänger eingeführt wurde. Ein Angreifer könnte also durch die Replay-Attacke möglicherweise die Log-Ergebnisse eines anderen Eigners empfangen, allerdings kann er sie nicht entschlüsseln. Somit hat er nicht mehr gewonnen, als wenn er generell die Kommunikationskanäle zu und von dem Aufbewahrer abhören würde.

7.2 Angriffe durch den Aufbewahrer

7.2.1 Versand falscher Schlüssel- oder Datenpakete

Schutzziele

Integrität und Verfügbarkeit

Motivation und Angriffsszenario

Es könnte sich um ein Versehen bzw. einen Implementierungs- oder Datenbankfehler handeln, wenn der Aufbewahrer die Anfragen von Verarbeitern mit falschen Schlüssel-

[248] Auch derjenige Verarbeiter, der die protokollierte Abfrage gestellt hat, kennt natürlich den Inhalt der Policy. Dieser hat aber kein Interesse an einer Logabfrage, da sie ihm nur seine eigenen ehemaligen Abfragen zeigen würde, und diese kennt er ohnehin.

oder Datenpaketen beantwortet. Dass der Aufbewahrer das System als Ganzes sabotieren möchte, ohne eigenen Gewinn davonzutragen, ist nicht anzunehmen. Das Protokoll beruht in der Tat darauf, dass Aufbewahrer und Zertifizierer ihren Verpflichtungen nachkommen, anderenfalls bricht der Datenaustausch ab. Aufbewahrer und Zertifizierer werden mutmaßlich für ihre Arbeit und ihre Speicherkapazität von den Nutzern bezahlt[249], sie haben eher kein Interesse daran, die Zuverlässigkeit des Systems in Frage zu stellen.

Verteidigung

Die Schlüssel- und Datenpakete, die beim Aufbewahrer liegen, sind durch die Dateneigner verschlüsselt. Dies geschah mit Schlüsseln, die neben dem Eigner nur dem beabsichtigten Verarbeiter (in Bezug auf die Datenpakete sowie in Szenario 1 auch die Schlüsselpakete) bzw. dem Zertifizierer (bezogen auf die Schlüsselpakete in Szenario 2) bekannt sind. Der Aufbewahrer kann also keine bewusste Manipulation durch Vertauschen von abgelegten Paketen beim Versand vornehmen. Sollte dennoch eine Verwechslung oder Vertauschung geschehen, sind verschiedene Fälle zu unterscheiden:

Handelt es sich um ein **Datenpaket**, das der Aufbewahrer vertauscht, wird der Verarbeiter es nicht öffnen können, da er nicht über das korrespondierende Schlüsselpaket verfügt. Schlimmstenfalls erhält der Aufbewahrer also eine Beschwerde des Verarbeiters und muss den Versand erneut mit dem korrekten Paket durchführen. Handelt es sich bei dem vertauschten Paket um ein **Schlüsselpaket**, wird dies in Szenario 1 ebenfalls keine schwerwiegenden Konsequenzen haben. Die Schlüsselpakete sind mit den öffentlichen Schlüsseln der beabsichtigten Verarbeiter verschlüsselt und können folglich auch allein von diesen und keinesfalls von einem fehlerhaft versorgten Empfänger entschlüsselt werden.

Wird in Szenario 2 ein Schlüsselpaket vertauscht, scheitert die Übertragung ebenfalls. Bei der Umschlüsselung eines falschen Schlüsselpakets würde der Zertifizierer nach Erhalt des Pakets vom Aufbewahrer das One-Time-Pad anwenden, das er vom Eigner für die entsprechende Empfängergruppe erhalten hat. Stammt das Paket entweder von einem anderen Eigner und/oder ist für eine andere Empfängergruppe vorgesehen, führt die Umschlüsselung beim Zertifizierer zu einem Chiffrat, für das kein Empfänger das korrekte One-Time-Pad besitzt, um es wieder zu entschlüsseln.

In allen geschilderten Fällen würde also die Vertauschung der Pakete zwar zu Beschwerden der Verarbeiter und gegebenenfalls Diskreditierung des Systems führen, nicht jedoch zum Kompromittieren der zu schützenden Daten.

[249] Oder sie sind Non-Profit-Organisationen, die der Aufrechterhaltung des Systems verpflichtet sind.

7.2.2 Ändern der Zuweisung von Policies zu Daten

Schutzziele

Integrität und Verfügbarkeit

Motivation und Angriffsszenario

Der Aufbewahrer erhält von einem Eigner A_1 gemäß Protokoll Datenpakete, die zum einen die verschlüsselten Daten $E_{1j}(M_{1l})$ bzw. $E_{1Gj}(M_{1l})$ und die ebenfalls verschlüsselten Policies $E_{1j}(Y_l, J_k) = P_{1jlk}$ bzw. P_{1Gjlk} enthalten. Beide sind für ihn nicht zu entschlüsseln. Er verfügt weder über den symmetrischen Datenschlüssel noch über den privaten Schlüssel des Verarbeiters B_j bzw. der Empfängergruppe B_{Gj}, mit dem er sich den Schlüssel aus dem Schlüsselpaket beschaffen könnte.

Nun könnte jedoch ein arglistiger B_2 durch Bestechung des Aufbewahrers versuchen, an Daten zu gelangen, die für B_1 hinterlegt wurden. Er könnte den Aufbewahrer dazu bringen, im Datenpaket dem Datum $E_{11}(M_{1l})$ durch Vertauschen die Policy P_{12lk} zuzuweisen. Dieses kann er selbst erzeugen, sofern A_1 auch für ihn Daten beim Aufbewahrer hinterlegt hat. Er muss dann nur eine beliebige unauffällige Existenzanfrage stellen, um E_{12} zu ermitteln. Hiermit verschlüsselt er die die von ihm vermutete Kombination aus Datentyp Y_l und Verarbeitungszweck J_k um eine gültige Policy P_{12lk} zu erhalten.

Verteidigung

Nach Vertauschung durch den Aufbewahrer würde diese Policy jedoch einem Datum zugeordnet sein, von dem weder der Aufbewahrer noch B_2 sagen können, ob es dem definierten Datentyp und Zweck entspricht. Noch ist es keinem von beiden möglich, den Inhalt zu entschlüsseln, da keiner von beiden über den privaten Schlüssel des B_1 verfügt, mit dem der zur Entschlüsselung des echten Datensatzes notwendige symmetrische Schlüssel K_{11} ermittelt werden könnte.

Eine Aufhebung der „Policy Stickyness“ ist also dem Aufbewahrer möglich, jedoch erhalten weder er noch ein anderer Beteiligter Vorteile daraus.

7.2.3 Leugnen von Datenverlust oder -verfälschung

Schutzziele

Integrität und Verfügbarkeit

Motivation und Angriffsszenario

Wiederum soll nicht unterstellt werden, der Aufbewahrer würde böswillig Daten vernichten. Schließlich ist das zuverlässige Speichern und protokollierte Weitergeben von Daten nach Prüfung der Policy seine Existenzberechtigung. Sind dem Aufbewahrer jedoch Datensätze abhanden gekommen oder sind Datensätze wegen mangelhafter

Sicherungsmaßnahmen unlesbar geworden, könnte er einen Anreiz haben, dies zu vertuschen. Gegebenenfalls drohen ihm Vertragsstrafen, die er vermeiden möchte. Auch sein Ruf als verlässlicher Aufbewahrer könnte bei Bekanntwerden eines Datenverlustes leiden.

Verteidigung

Grundsätzlich scheint es für Dateneigner sinnvoll zu sein, sich die initiale Hinterlegung ihrer Daten und alle späteren Änderungen durch den Aufbewahrer mittels dessen digitaler Signatur quittieren zu lassen. Auf diesem Weg haben sie die Beweise in der Hand, falls zu einem späteren Zeitpunkt ein Verarbeiter Sanktionen einleiten möchte, weil Daten fehlen. Der Eigner kann dann die signierten Quittungen vorweisen und damit die Sanktionen auf den Aufbewahrer lenken. Setzt man Hashwerte über die übermittelten Datenpakete ein, die vom Aufbewahrer signiert und dann veröffentlicht werden, bleibt hierbei auch die Anonymität des Dateneigners gegenüber dem Aufbewahrer gewahrt.

Das Aufdecken der Empfangsquittungen ist allerdings nur eine nachgelagerte Maßnahme, die bei Rechtsstreitigkeiten sinnvoll sein kann. Geht es jedoch wie im Beispiel der Notfall-Mediziner darum, dass von dem Vorhandensein der Daten zu einem bestimmten Zeitpunkt ein unwiederbringliches Gut abhängt, wie etwa die Gesundheit des Betroffenen, greift die nachträgliche juristische Nachweisbarkeit eines Fehlverhaltens des Aufbewahrers offensichtlich zu kurz. An dieser Stelle muss über Maßnahmen diskutiert werden, die einen regelmäßigen Nachweis der zuverlässigen Datenverwahrung erbringen.

Shah et al. schlagen für das allgemeine Problem der Aufbewahrung von Daten bei Dritten, im Zusammenhang mit der Kontrolle der hinterlegten Daten auf Vorhandensein und Integrität, ein Protokoll vor, das einen Auditor einführt, der diese durch Zero-Knowledge-Beweise – also ohne etwas von den Inhalten zu erfahren – prüfen kann.[250]

7.2.4 Verfälschen oder Löschen der Übermittlungsprotokolle

Schutzziele

Integrität und Verbindlichkeit

Motivation und Angriffsszenario

Der Versuch, die Abfrageprotokolle zu manipulieren, mag aus ähnlichem Hintergrund geschehen, wie das (versehentliche) Löschen oder Verfälschen der Datenpakete. Der Aufbewahrer könnte versuchen, eigene Fehler, die seine Vertrauenswürdigkeit und damit sein Geschäftsmodell negativ beeinflussen, zu vertuschen.

[250] vgl. [ShSB08]

Verteidigung

Wie auch bei der Verfälschung der abgelegten Datenpakete ist es nicht zu verhindern, dass ein Aufbewahrer seiner Verantwortlichkeit nicht nachkommt und Protokolldaten verändert. Dabei ist eine manipulative Veränderung, das heißt mit dem Ziel, eine bestimmte Reaktion herbeizuführen, nur begrenzt möglich. Der Aufbewahrer kann aus eigener Kraft keine sinnvollen Policy-Pakete erstellen, mit denen er eine realistische Verarbeiter-Abfrage vorspiegeln könnte, die nicht stattgefunden hat. Dazu fehlt ihm die Möglichkeit, ein gültiges Policy-Chiffrat $P_{ijlk} = E_{ij}(Y_l, J_k)$ bzw. $P_{iGjlk} = E_{iGj}(Y_l, J_k)$ zu erzeugen. Schließlich verfügt er nicht über dafür notwendigen Schlüssel K_{ij} bzw. K_{iGj}. Realistisch kann er also nur die Zeitstempel verfälschen, mit denen er die Abfragen der Verarbeiter versieht, ganze Protokoll-Sätze löschen oder eine bereits vorliegende Abfrage ein zweites Mal zu einem anderen Zeitpunkt in das Protokoll einfügen. Es scheint jedoch hinreichend schwer für den Aufbewahrer zu sein, aus diesen Arten der Manipulation eigenen Nutzen zu ziehen. Wiederum könnte er also nur das Vertrauen in das Protokoll schädigen, was er ebenso durch generelles Abschalten seiner eigenen Dienste erreichen könnte und damit sein Geschäftsmodell hinfällig werden ließe.

Nichts desto trotz könnte ein zusätzlicher Sicherheitsmechanismus in das Protokoll eingeführt werden, der das Verfälschen, das Löschen und auch das versehentliche Verlieren von Daten, Schlüsseln und Protokollen erschwert. Dazu könnte beispielsweise ein zweiter Aufbewahrer eingeführt werden, der unabhängig vom ersten operiert, und der in vollständiger Redundanz zum ersten Aufbewahrer arbeitet. Beide Aufbewahrer erhielten alle Pakete der Eigner sowie alle Anfragen und würden die Anfragen auch parallel bearbeiten. Ein Verarbeiter würde im Normalbetrieb immer zwei gleichlautende Antworten auf seine Anfragen erhalten und könnte bei Abweichungen Alarm schlagen, das heißt die Nutzergemeinde des Systems über die „Ungereimtheiten“ informieren.

7.2.5 Brechen der Policy-Chiffrate

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Gegebenenfalls möchte der Aufbewahrer die Informationen darüber auswerten, welche Arten von Daten für welche Verarbeiter oder Empfängergruppen hinterlegt wurden. Er könnte daraus vermarktbare Einsichten über die Aktivitäten der Verarbeiter gewinnen.

Verteidigung

Dazu müsste der Aufbewahrer ermitteln, welche Policies, das heißt Kombinationen von Datentyp, Verarbeitungszweck und Empfänger(gruppe) in seinen Datenbanken liegen. Dies könnte er tun, wenn die Datensätze direkt mit dem öffentlichen Schlüssel der Verarbeiter verschlüsselt wären. Denn dann könnte er sich ebenfalls die öffentlichen

Schlüssel besorgen und jede mögliche Kombination der (per Definition bekannten) Datentypen Y_l und Verarbeitungszwecke J_k damit verschlüsseln sowie mit den pro Eigner hinterlegten verschlüsselten Datensatz vergleichen. Hier gelänge ein Brute Force-Angriff, da die möglichen Kombinationen von Typ und Zweck mit $l \cdot k$ Möglichkeiten übersichtlich sind und exhaustiv getestet werden können.

Da die Datensätze aber mit einem symmetrischen Datenschlüssel verschlüsselt und dieser dann mit dem öffentlichen Schlüssel des Verarbeiters verschlüsselt wird, kann der Aufbewahrer die Methode nicht anwenden. Brute Force wird bei hinreichend großer Länge des symmetrischen Schlüssels nicht zum Erfolg führen.

7.2.6 Brechen der Datenschlüssel

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Der symmetrische Datenschlüssel ist das Kernelement im Protokoll zur Sicherung der personenbezogenen Daten und auch zur verbindlichen Empfängerzuordnung aller Abfragen. Gelingt es dem Aufbewahrer, diesen Schlüssel zu ermitteln, kann er sowohl an die Daten der Eigner gelangen, als auch manipulierte Abfragen, Policies und Datenpakete in seinem Sinne generieren.

Durch Brute-Force ist der Schlüssel nicht zu brechen, wenn er über eine ausreichende Länge verfügt. Mehr Erfolg verspricht ein Known-Plaintext-Angriff.

Verteidigung

Dieser Angriff kann gelingen, da die Kombinationen der bekannten – da veröffentlichten – Datentypen Y_l und Verarbeitungszwecke J_k mit $l \cdot k$ Möglichkeiten getestet werden können. Ebenso können einige der korrespondierenden Datensätze erraten werden, z.B. sind die möglichen Werte für einen Datentyp „Religion“ überschaubar. T könnte sich also zunächst die hinterlegten Policy-Datensätze nacheinander vornehmen, und sie gegen eine angenommen häufige Klartextkombination, z.B. „Religion / Lohnsteuerabrechnung / Katholisch“ testen. Der Angriff wird dann mit einiger Wahrscheinlichkeit zum Erfolg führen, wenn der verwendete Kryptoalgorithmus für die symmetrische Datenverschlüsselung anfällig gegen Known-Plaintext-Angriffe ist.

Traut man dem Algorithmus alleine nicht die notwendige Stabilität gegen diese Art des Angriffs zu, so erhöht sich gegebenenfalls der Schutz durch Blenden der Policies und Datenpakete mit Zufallszahlen, die zusammen mit dem symmetrischen Schlüssel durch den öffentlichen Schlüssel des Verarbeiters verschlüsselt werden[251]. Der Verarbeiter würde dann nach Erhalt des Datenschlüssels und damit auch der Zufallszahl(en) den

[251] Das Kryptosystem ist dann indeterministisch in Bezug auf die Klartexte.

zusätzlichen Schritt im Protokoll aufnehmen müssen, die passende Zufallszahl in seine für die Abfragen generierten Policy-Chiffrate einzuarbeiten bzw. die Zufallszahlen-Maskierung aus den zurück erhaltenen Datenpaketen wieder heraus zu rechnen.

Auf jeden Fall sollte ein Aufbewahrer nicht an der Erstellung der allgemein gültigen Liste der Datentypen und Verwendungszwecke mitwirken. Sonst hätte er die Möglichkeit, über die Known-Plaintext-Attacke hinaus sogar eine Chosen-Plaintext-Attacke durchzuführen, die mindestens so effizient ist.

7.2.7 Ableiten von Informationen aus Verarbeiter-Abfragen

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Ist dem Aufbewahrer daran gelegen, möglichst viele Informationen über die Beziehungen zwischen Dateneignern und Verarbeitern sowie die ausgetauschten Daten zu erfahren, weil er beispielsweise durch Data Mining vermarktbare Informationen daraus ziehen möchte, könnte er sich durch die Analyse der Anfrage- und Antwortmuster Erfolge versprechen.

Aus den Anfragen der Verarbeiter auf Existenz von bestimmten Policies (also Kombinationen von Datentyp und Verarbeitungszweck) und der entsprechenden positiven oder negativen Rückmeldungen könnte der Aufbewahrer Schlüsse über die Art der hinterlegten Daten ziehen.

Verteidigung

Um die Qualität des Erkenntnisgewinns für den Aufbewahrer zu beurteilen, muss man die ihm verfügbaren Daten betrachten: Er kennt den Klartext der Policies nicht, und je nach Ausgestaltung der initialen Datenablage ebenso wenig die Dateneigner, die Urheber der Policies und Datenpakete sind. In Szenario 1 kann er beobachten, dass und wie oft ein bestimmter Datenverarbeiter Existenz- und Inhaltsabfragen stellt. Informiert er sich über die Funktion des Verarbeiters und bringt das in Korrelation mit der Erfolgsquote der Abfragen, kann er möglicherweise daraus schließen, wie der Verarbeiter arbeitet. Was er jedoch nicht in Erfahrung bringen kann, sind personenbezogene Daten eines Eigners oder die Verbindungen, die Eigner zu bestimmten Verarbeitern haben[252].

In Szenario 2 ist es für den Aufbewahrer zwar einfacher, aufgrund der Zugehörigkeit eines Anfragenden zu einer bestimmten Empfängergruppe zu identifizieren, was derjenige mit den Daten zu tun beabsichtigt, über die Daten selbst oder die Beziehungen zu bestimmten Eignern erfährt er jedoch nach wie vor nichts.

[252] Diese Annahme gilt natürlich erst dann, wenn der Aufbewahrer nicht nur für einen einzelnen Dateneigner tätig ist.

Sollte es dennoch sensibel sein, dass der Aufbewahrer aus der Häufigkeit und der Erfolgsquote von Abfragen Schlüsse zieht, könnte der Abfrage-Client, der durch die Verarbeiter genutzt wird, so gestaltet werden, dass er regelmäßig Dummy-Abfragen schickt und so das tatsächliche Aufkommen und die Erfolgsquote der realen Abfragen verschleiert.

7.2.8 Ableiten von Informationen aus den abgelegten Datenmengen

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Hat sich ein Aufbewahrer beispielsweise auf die Datenspeicherung im Bereich des Gesundheitswesens spezialisiert, würde ihm möglicherweise schon die Information, dass ein bestimmter Eigner eine besonders große Datenmenge bei ihm abgelegt hat, verwertbare Rückschlüsse erlauben. Man könnte aus einer großen Datenmenge folgern, dass die Krankengeschichte des Eigners besonders umfangreich ist, und damit eine Einstellung dieser Person für einen potentiellen Arbeitgeber mit einem erhöhten Ausfallrisiko versehen wäre.

Verteidigung

Der wirkungsvollste Schutz gegen diesen Angriff ist zunächst, dass das Protokoll eine anonyme Ablage der Daten erlaubt. So kann der Aufbewahrer zwar messen, wie viel Daten bei ihm abgelegt werden, aber nicht woher sie stammen. Dies macht die Information unnütz für den Zweck der vorgenannten Auswertung. Auch das Warten darauf, dass ein Eigner seine Daten früher oder später aktualisieren muss und damit seine Identität preisgäbe, ist sinnlos. Denn ein Dateneigner authentisiert sich nicht durch seine Identität als berechtigt ein bestimmtes Datenpaket zu ändern, sondern durch die Kombination aus Policy und Zufallszahl (bzw. deren Hashwert), mit der er das korrekte Datenpaket adressiert und die nur er kennen kann. Ebenso wenig wie er die Identität eines anonymen Dateneigners aufdecken kann, kann der Aufbewahrer mehrere Kontakte miteinander korrelieren. Das bedeutet, er weiß nicht, ob zwei Datenaktualisierungen von demselben Eigner kommen oder von unterschiedlichen Eignern stammen.

Optional kann zur zusätzlichen Verschleierung der tatsächlich abgelegten Datenmengen natürlich auch die Verwendung von Dummy-Einträgen dienen, die durch die Paketierungsapplikation des Eigners eingestreut werden könnten. Diese Dummy-Pakete wären solchermaßen zu gestalten, dass sie keiner echten Policy entsprechen. Sie wären also niemals Inhalt einer Abfrage – oder höchstens einer Dummy-Abfrage gemäß Abschnitt 7.2.7, wenn man die verschiedenen Schutzmechanismen miteinander verknüpfen möchte. Die wahre Größe abgelegter Datenpakete könnte man zudem durch

eine Normierung der Ablagegrößen verschleiern, kleine Datenmengen würden durch Zufallswerte aufgefüllt, große Mengen in mehrere Pakete aufgeteilt.

7.2.9 Man-in-the-Middle-Angriff bei Schlüsselabfrage

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Auch wenn der Aufbewahrer nicht an die hinterlegten Daten der Dateneigner gelangt, könnte er doch sowohl den Eignern als auch den Verarbeitern massiv schaden, wenn es ihm gelänge, falsche Daten in das Protokoll einzuschleusen. Er könnte dem Verarbeiter einen vermeintlich vom Eigner erzeugten Datensatz übermitteln, den dieser für glaubhaft hält, obwohl der Eigner diese Daten gar nicht oder in anderer Form abgelegt hat. Um diese Täuschung durchzuführen, greift der Aufbewahrer auf einen Man-in-the-Middle-Angriff zurück.

Verteidigung

Der Aufbewahrer könnte den vom Eigner erzeugten symmetrischen und dann mit dem öffentlichen Schlüssel des Verarbeiters verschlüsselten Datenschüssel K_{ij} durch einen von ihm selbst erzeugten Schlüssel K_{Tj} ersetzen. Diesen Schlüssel würde er mit dem öffentlichen Schlüssel des Verarbeiters verschlüsseln und diesem bei der ersten Schlüsselabfrage schicken. Fortan hätte er die Macht über alle weiteren Abfragen des betreffenden Verarbeiters, da dieser nun seine Kommunikation mit dem Schlüssel K_{Tj} chiffrieren würde, den der Aufbewahrer sehr wohl kennt. Er könnte diese Anfragen mit selbst erstellten Datenpaketen beantworten, die er wiederum mit K_{Tj} verschlüsselt. Der Empfänger würde annehmen, dass die Daten vom Eigner stammen und entsprechend reagieren. Der Aufbewahrer käme so zwar nicht an die vom Eigner eigentlich hinterlegten Informationen, aber das Vertrauens- oder Geschäftsverhältnis zwischen den beiden Parteien hätte er dauerhaft beeinflusst oder gestört.

Abgewendet wird diese Gefahr dadurch, dass gemäß Protokoll jedes Schlüsselpaket durch den Eigner mit seinem eigenen privaten Schlüssel signiert wird. Ein Verarbeiter kann nach der Antwort auf seine Schlüsselabfrage sofort prüfen, ob der erhaltene Schlüssel tatsächlich auf den korrekten Eigner als Quelle zurückzuführen ist und nicht verfälscht oder ersetzt wurde. Der Aufbewahrer kann ohne Kenntnis des privaten Schlüssels des Eigners diese Signatur nicht imitieren. Somit läuft sein Angriff ins Leere, weil die Verarbeiter an der fehlenden Signatur jederzeit den Fälschungsversuch erkennen können.

7.2.10 Auftreten als Verarbeiter

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Der Aufbewahrer hat möglicherweise Interesse daran, die Daten von einem oder mehreren Eignern zu erfahren. Er könnte sich nun selbst als Datenverarbeiter in das Protokoll aufnehmen, und Schlüssel-, Existenz- und Inhaltsabfragen bezüglich verschiedener Eigner an sich selbst stellen, bis er einen Treffer erzielt.[253]

Verteidigung

In Szenario 1 wird dieser Angriff dadurch verhindert, dass die Eigner selbst auswählen, für wen sie Daten vorhalten möchten. Sie besorgen sich die öffentlichen Schlüssel der gewählten Verarbeiter und generieren individuelle Schlüsselpakete für diese. Der Aufbewahrer müsste also die Identität oder zumindest den privaten Schlüssel eines potentiell korrekten Verarbeiters stehlen, um diesen Angriff mit Erfolg durchführen zu können.

In Szenario 2 wird der Angriff dadurch verhindert, dass der Zertifizierungsinstanz Beweise vorgelegt werden müssen, die die Aufnahme eines Verarbeiters in Empfängergruppen rechtfertigen. Ob es dem Aufbewahrer durch Täuschung gelingt, in eine Gruppe aufgenommen zu werden, hängt also davon ab, wie robust die Prüfungen des Zertifizierers gegenüber gefälschten Nachweisen sind.

[253] Wegen der Anonymität der Eigner bei der Datenablage und Aktualisierung weiß er ja zunächst nicht, für wen sich eine Abfrage lohnt. Also muss er ihm verfügbare Adressverzeichnisse durchgehen, bis er einen Eintrag gefunden hat, für den er Daten in seinem Speicher liegen hat.

7.3 Angriffe durch den Zertifizierer

Die in diesem Abschnitt diskutierten Angriffe beziehen sich nur auf Szenario 2 des vorgestellten Protokolls, Szenario 1 verfügt schließlich nicht über die Rolle einer Zertifizierungsinstanz.

7.3.1 Selbstzertifizierung als Empfängergruppen-Mitglied

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Der Zertifizierer hat die Macht, Verarbeiter den Empfängergruppen hinzuzufügen. Nun könnte er sich selbst einige Identitäten zulegen, die er für die verschiedenen Gruppen zertifiziert. Unter deren Flagge könnte er dann die Schlüsselabfragen und später die Existenz- und Inhaltsabfragen beim Aufbewahrer stellen. Der Aufbewahrer prüft die Rechtmäßigkeit der Anfragen anhand der vom Zertifizierer veröffentlichten Mitgliedslisten und übermittelt die angeforderten Datenpakete, sofern er der Täuschung erliegt.

Verteidigung

Zu einem gewissen Maße besteht eine soziale Kontrolle über das Gebaren des Zertifizierers. Da er die Mitgliederlisten der von ihm verwalteten Empfängergruppen veröffentlicht, wird massiver und regelmäßiger Missbrauch seiner Stellung früher oder später durch aufmerksame Beobachter aufgedeckt werden. Nutzt er seine Position jedoch so geschickt, dass er nur in wenigen Fällen die Mitgliederlisten manipuliert, dann sofort die entsprechenden Anfragen stellt und danach unmittelbar die Empfängergruppen wieder bereinigt, wird er durch unregelmäßige Stichproben kaum zu entlarven sein. Sollte die Zertifizierungsinstanz also nicht eine Institution sein, über deren Vertrauenswürdigkeit allgemeiner Konsens herrscht, müsste sie einer weiteren Kontrollinstanz unterworfen werden. Hierfür würde sich etwa eine Gruppe von Dateneignern anbieten, die sich alle Änderungen an Mitgliederlisten inklusive der eingereichten Legitimationen vorlegen lassen, bevor die Mitgliedschaft veröffentlicht und damit dem Aufbewahrer bekannt gemacht wird.

Alternativ ließe sich das Verfahren so anpassen, dass es mehrere Zertifizierer gäbe und zur Aufnahme eines Bewerbers in eine Empfängergruppe immer alle oder zumindest mehrere Zertifizierer zustimmen müssten. Ein Verarbeiter, der als Mitglied einer Empfängergruppe veröffentlicht wäre, würde nur dann von den anderen Parteien des Verfahrens akzeptiert werden (und damit vom Aufbewahrer mit einem entsprechenden One-Time-Pad versorgt werden), wenn sein Kennung von einer festgelegten Mindestanzahl von Zertifizierern signiert wurde.

7.3.2 Dechiffrieren der Datenschlüssel

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

In Szenario 2 ist die Anwendung der beiden verschiedenen One-Time-Pads, OTP_{iGj} und OTP_{Tj} notwendig, um den Datenschlüssel eines Eigners für eine Empfängergruppe zu erhalten. Im Verlauf des Protokolls werden die One-Time-Pads von den beteiligten Parteien im Wechsel angewandt, wobei jede Partei immer exakt eines der beiden Pads kennt (vgl. Abbildung 16 in Abschnitt 5.2.3.3). Wer die Kontrolle über beide Pads hätte, könnte während des Schlüsselaustauschs den Datenschlüssel abfangen und vollständig dechiffrieren.

Verteidigung

Der Zertifizierer erhält gemäß dem Protokoll die Kenntnis von OTP_{iGj}. Das komplementäre One-Time-Pad, also OTP_{Tj} wird vom Aufbewahrer erstellt und an den anfragenden Verarbeiter geschickt. Davor wird es mit dessen öffentlichem Schlüssel verschlüsselt. Selbst wenn der Zertifizierer also den Datenverkehr zwischen Aufbewahrer und Verarbeiter abhören sollte, könnte er mangels des privaten Schlüssels von B_j das One-Time-Pad nicht dechiffrieren und somit auch nicht auf das Chiffrat des Datenschlüssels anwenden. Die Verschlüsselung der Kommunikation mit den jeweiligen öffentlichen Schlüsseln der Empfänger verhindert im Übrigen auch gegenüber jeder anderen Partei, dass die per One-Time-Pad verschlüsselten und dann mit dem öffentlichen Schlüssel erneut chiffrierten Kommunikationen durch Kombinieren mehrerer Nachrichten – wie es bei der alleinigen Anwendung des One-Time-Pad der Fall wäre – aufgedeckt werden können.

7.3.3 Dechiffrieren der Datenschlüssel nach Kompromittierung des Aufbewahrers

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Analog zu Abschnitt 7.3.2.

Gelänge es dem Zertifizierer auf irgendeine Weise, an das OTP_{Tj} zu gelangen, könnte er es in Kombination mit dem ihm ohnehin verfügbaren OTP_{iGj} nutzen, um die Schlüsselpakete zwischen A_i und den Mitgliedern von B_{Gj} zu dechiffrieren. Dasselbe gilt für die Situation, in welcher der Aufbewahrer Kenntnis von den OTP_{iGj} erlangt, die beim Zertifizierer lagern. Dies wäre unter Umständen noch riskanter, da der Aufbewahrer auch über die

Schlüsselpakete verfügt. Er müsste also nicht einmal eine Kommunikation abhören, um als möglicher Angreifer auf die Datenschlüssel zum Erfolg zu kommen.

Verteidigung

In der Tat ist der missbräuchliche Datenaustausch zwischen Aufbewahrer und Zertifizierer, unabhängig davon ob er durch Kooperation oder Angriffe zustande kommt, die größte Schwachstelle im Verfahren PDG/v. Ihr könnte man jedoch durch Rückgriff auf Techniken der Mehrparteienberechnung [254] begegnen und anstelle *eines* Zertifizierers mehrere seiner Art einsetzen, die zusammen wirken müssen, um das OTP_{Tj} vollständig anzuwenden. Dass es sich bei den zu schützenden Objekten um One-Time-Pads handelt, macht aufgrund deren günstiger Eigenschaften, insbesondere der Kommutativität ihrer Anwendung, eine Verteilung auf mehrere Parteien sehr einfach.

Die Aufgaben des Zertifizierers sind neben der Verwaltung der Empfängergruppen[255] im Verfahren:

- die Entgegennahme und Aufbewahrung der Gruppen-OTPs,
- deren Anwendung im Rahmen der ersten OTP-Entschlüsselung und
- die Weitergabe des verbleibenden Chiffrats an den beabsichtigten Empfänger.

Diese Aufgaben seien also auf mehrere Zertifizierungsinstanzen Z_q, mit $q \in \{1, 2, \ldots, u\}$ und u als Anzahl der Zertifizierungsinstanzen zu verteilen. Dann ist es die Aufgabe des Dateneigners, anstelle eines OTP_{iGj} mehrere, nämlich u One-Time-Pads zu erstellen, ein OTP_{iGjq} für jeden der Teil-Zertifizierer. Diese OTP_{iGj1} bis OTP_{iGju} lässt er zufällig generieren. Anschließend verknüpft er alle OTPs nacheinander mit bitweisem XOR und erhält sein persönliches OTP_{iGj}, das er mit niemandem teilt. Er nutzt es nur zur ersten Verschlüsselung des Schlüsselpakets.

Sodann verschickt er an jeden der Z_q willkürlich ausgewählt eines der generierten OTP_{iGjq}, jeder Teil-Zertifizierer erhält ein anderes OTP_{iGjq} und bewahrt es als sein Teilwissen des OTP_{iGj} auf. Muss nun für ein Schlüsselpaket die erste OTP-Entschlüsselung durchgeführt werden, so sendet der Aufbewahrer sein doppelt verschlüsseltes Chiffrat $OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))$ an jede der Zertifizierungsinstanzen. Diese wenden jeweils ihren OTP_{iGjq} an und senden das Resultat an den Verarbeiter B_j. Sobald der Verarbeiter u Nachrichten empfangen hat, muss er sie nur alle nacheinander bitweise per XOR ($\oplus$) verknüpfen und erhält so das nur noch einfach chiffrierte Schlüsselpaket.

Im Beispiel mit $u = 3$ entsteht das OTP_{iGj} als:

$OTP_{iGj} = OTP_{iGj3}(OTP_{iGj2}(OTP_{iGj1}))$, oder anders notiert:

[254] vgl. [DuAt01], [OnLZ09]

[255] Auch die Verwaltung der Empfängergruppen kann auf mehrere Zertifizierer-Instanzen aufgeteilt werden. Durch den Einsatz von Gruppensignaturen (vgl. [ChHe91], [BeMW03], [KiTY04], [KiYu05]) können verschiedene Zertifizierer die Empfänger so als Gruppenmitglieder beglaubigen, als wären sie von einer zentralen Instanz zertifiziert.

$$OTP_{iGj} = OTP_{iGj1} \oplus OTP_{iGj2} \oplus OTP_{iGj3}$$

Dann erhält der Verarbeiter:

1. $OTP_{iGj1}(OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))) =$
   $= OTP_{iGj1} \oplus OTP_{Tj} \oplus OTP_{iGj} \oplus (S_i(K_{iGj})) =$
   $= OTP_{iGj1} \oplus OTP_{Tj} \oplus OTP_{iGj1} \oplus OTP_{iGj2} \oplus OTP_{iGj3} \oplus (S_i(K_{iGj}))$
2. $OTP_{iGj2}(OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))) =$
   $= OTP_{iGj2} \oplus OTP_{Tj} \oplus OTP_{iGj} \oplus (S_i(K_{iGj})) =$
   $= OTP_{iGj2} \oplus OTP_{Tj} \oplus OTP_{iGj1} \oplus OTP_{iGj2} \oplus OTP_{iGj3} \oplus (S_i(K_{iGj}))$
3. $OTP_{iGj3}(OTP_{Tj}(OTP_{iGj}(S_i(K_{iGj})))) =$
   $= OTP_{iGj3} \oplus OTP_{Tj} \oplus OTP_{iGj} \oplus (S_i(K_{iGj})) =$
   $= OTP_{iGj3} \oplus OTP_{Tj} \oplus OTP_{iGj1} \oplus OTP_{iGj2} \oplus OTP_{iGj3} \oplus (S_i(K_{iGj}))$

Verknüpft er sie, erhält er aufgrund der Kommutativität des XOR und seiner Eigenschaft, bei zweimaliger Anwendung wieder den Klartext hervorzubringen:

$$OTP_{iGj1} \oplus OTP_{Tj} \oplus OTP_{iGj1} \oplus OTP_{iGj2} \oplus OTP_{iGj3} \oplus (S_i(K_{iGj})) \oplus OTP_{iGj2} \oplus OTP_{Tj} \oplus$$
$$OTP_{iGj1} \oplus OTP_{iGj2} \oplus OTP_{iGj3} \oplus (S_i(K_{iGj})) \oplus OTP_{iGj3} \oplus OTP_{Tj} \oplus OTP_{iGj1} \oplus OTP_{iGj2} \oplus$$
$$OTP_{iGj3} \oplus (S_i(K_{iGj})) =$$
$$= (OTP_{iGj1} \oplus OTP_{iGj1} \oplus OTP_{iGj1} \oplus OTP_{iGj1}) \oplus (OTP_{iGj2} \oplus OTP_{iGj2} \oplus OTP_{iGj2} \oplus OTP_{iGj2})$$
$$\oplus (OTP_{iGj3} \oplus OTP_{iGj3} \oplus OTP_{iGj3} \oplus OTP_{iGj3}) \oplus (OTP_{Tj} \oplus OTP_{Tj} \oplus OTP_{Tj}) \oplus (S_i(K_{iGj})) \oplus$$
$$(S_i(K_{iGj})) \oplus (S_i(K_{iGj})) =$$
$$= OTP_{Tj} \oplus (S_i(K_{iGj})) =$$
$$= OTP_{Tj}(S_i(K_{iGj}))$$

Auf dieses wendet er das Empfänger-OTP an, das er gemäß der geschilderten Schritte im Verfahren vom Aufbewahrer erhalten hat. Im Resultat erhält er das dechiffrierte Schlüsselpaket.

Man sieht, dass das Verfahren unter dieser Erweiterung arbeitet, ohne dass ein Zertifizierer das vollständige Gruppen-OTP kennt. Um nun noch das Verfahren auf Seiten der One-Time-Pads zu kompromittieren, wäre die Zusammenarbeit der u+1 Parteien (alle Teil-Zertifizierer und der Aufbewahrer) nötig, was den Aufwand und das Risiko eines Angriffs signifikant erhöhen würde.

Alternativ kann Shamirs Secret Sharing-Verfahren Anwendung finden[256]. Dieses erlaubt die Abbildung eines (g, u)-Schwellwertschemas, bei dem von u Teil-Geheimnissen mindestens g ($u \geq g$) aufgedeckt werden müssen, um das Geheimnis zu enthüllen. Dabei

[256] vgl. [Sham79], [BeSW04]

wählt der Dateneigner ein Polynom vom Grad g-1, wobei das Geheimnis OTP_{iGj} der konstante Term (das Absolutglied) ist und die restlichen Koeffizienten zufällig gewählt sind:

$$f(x) = a_1 \cdot x + a_2 \cdot x^2 + \ldots + a_{g-1} \cdot x^{g-1} + OTP_{iGj}\,.$$

Er berechnet u Stellen des Polynoms und verteilt sie als die Teilgeheimnisse an die Zertifizierungsinstanzen Z_q mit $q \in \{1, 2, \ldots, u\}$. Ein Verarbeiter kann das OTP_{iGj} erst dann ermitteln, wenn er von den Zertifizierern mindestens g Teilgeheimnisse erhalten hat. Denn dann kann er durch Interpolation das Polynom ermitteln und somit auch dessen Absolutglied OTP_{iGj} erhalten. Auf diese Weise wird verhindert, dass ein Zertifizierer alleine nach einer Kompromittierung des Aufbewahrers das Verfahren missbräuchlich nutzen kann, das System verkraftet aber den Ausfall von (u-g) Zertifizierungsinstanzen.

7.4 Angriffe durch Datenverarbeiter

7.4.1 Leugnen einer Datenabfrage

Schutzziel

Verbindlichkeit

Motivation und Angriffsszenario

Sollte sich ein Dateneigner zu einem beliebigen Zeitpunkt darüber beschweren, dass ein Verarbeiter seine Daten abgefragt hat, obwohl aus seiner Perspektive der rechtfertigende Verarbeitungszweck nicht eingetreten war, würde der beschuldigte Verarbeiter vermutlich gerne abstreiten, die Daten abgefragt zu haben. So könnte er einer entsprechenden Klage des Dateneigners entgegenwirken.

Verteidigung

Der Aufbewahrer führt zu jeder Existenz- und Inhaltsabfrage einen Eintrag in seinem Protokollsystem. Auch wenn der Aufbewahrer selbst die Protokolleinträge nicht entziffern kann (die angefragten Policies sind ja mit dem Datenschlüssel verschlüsselt, den er nicht im Klartext kennt), so kann jeder Eigner gegenüber einer jeden anderen Partei den Inhalt eines ihn betreffenden Protokolleintrags offenbaren und damit nachprüfbar machen. Er würde beispielsweise gegenüber einem Gericht seinen Datenschlüssel offenlegen, mit dem die fragliche Policy verschlüsselt wurde. Das Gericht könnte dann zum einen die Signatur des Abfragenden prüfen, die durch den Aufbewahrer ja mit der Abfrage im Protokoll gespeichert wurde, und zum anderen nach Kenntnis des Datenschlüssels die Policy prüfen und feststellen, ob die Abfrage aufgrund der Umstände und des Informationsbedarfs des Verarbeiters gerechtfertigt war. Ein Abstreiten seitens des Verarbeiters wäre aufgrund seiner Signatur an der Abfrage fruchtlos.

7.4.2 Abfragen aller möglichen Datensätze eines Eigners

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Prinzipiell kann ein Verarbeiter alle Datensätze abfragen, die ein Eigner für ihn hinterlegt hat. Er muss dazu nach erfolgreicher Schlüsselabfrage nur nacheinander Abfrage-Policies für die einzelnen Datentypen und Verarbeitungszwecke formulieren und an den Aufbewahrer stellen. Dieser kann in Unkenntnis des Inhalts der Policy-Chiffrate keine Qualifizierung in gerechtfertigte und ungerechtfertigte Abfragen vornehmen. Er wird die Datenpakete übermitteln, sofern sich Übereinstimmungen in den Policy-Chiffraten finden.

Verteidigung

Grundsätzlich scheint es hier zunächst ratsam zu fordern, dass Dateneigner sorgfältig abwägen, welche Datentypen sie für welche Verarbeiter zur Verfügung stellen wollen. Denn wenn eine ungerechtfertigte Inhaltsabfrage erst stattgefunden hat und bedient wurde, sind die Daten im Umlauf. Dem Eigner bleibt der juristische Weg, über die Protokolle des Aufbewahrers Gerechtigkeit zu erhalten, aber das wird nicht in jedem Fall den Verlust durch die Preisgabe der personenbezogenen Daten ausgleichen. Man könnte in das Protokoll einige heuristische Schutzmaßnahmen einbauen, wie etwa die Regel, dass ein Aufbewahrer nur eine festgelegte Anzahl von Abfragen eines Verarbeiters pro Tag zulässt. Die Abwägung des Nutzens gegen die dann mögliche Einschränkung eines gerechtfertigten Service-Levels ist anhand der Sensitivität der hinterlegten personenbezogenen Daten vorzunehmen.

7.4.3 Missbrauch der erhaltenen Daten

Schutzziele

Vertraulichkeit und Zweckbindung

Motivation und Angriffsszenario

Jeder Empfänger kann potentiell in Versuchung gelangen, die personenbezogenen Daten, die er von einem Eigner erhalten hat, zu anderen Zwecken als dem ursprünglich vereinbarten einzusetzen. Dies kann insbesondere durch die Potentiale des Direktmarketings, d.h. einer zielgruppengenauen Ansprache der Betroffenen durch Werbepartner des Verarbeiters erfolgen [257]. Aber auch vermeintlich gemeinnützige

[257] Howard Beales belegt in einer Studie [Beal10], die erhöhte Wirksamkeit von zielgerichteter Online-Werbung gegenüber unspezifischer Werbung. Er befragte 12 Werbenetzwerke, die zusammen durchschnittlich 78% der US-amerikanischen Bevölkerung erreichen. Im Jahr 2009 führte streuverteilte Online-Werbung zu einer „Conversion Rate“ (ein Klick auf das Werbebanner führt zu einem Kauf) von 2,8%. Wurde die Werbung auf den Benutzer abgestimmt („Behavioral Targeting“) lag die Rate bei 6,8%.

Tätigkeiten wie etwa Veröffentlichungen über Wikileaks können eine Motivation für zweckfremde Datennutzung darstellen.

Verteidigung

Das hier vorgestellte Protokoll legt gerade deshalb seinen Fokus bei der Herausgabe von personenbezogenen Daten auf den letztmöglichen Moment nach sorgfältiger Prüfung der Legitimation, weil es anerkennt, dass einmal übergebene Daten immer missbräuchlich genutzt werden können. So ist es auch im vorgestellten System der Fall. Missbräuchliche Nutzung kann zwar zum Teil durch Auswertung der Protokolle nachgewiesen und entsprechend strafrechtlich verfolgt werden, aber verhindert werden kann sie nicht.

Möglichkeiten, die bestehen, um zumindest eine automatische unrechtmäßige Weitergabe von übertragenen Daten zu limitieren, würden voraussetzen, dass sich die am System teilnehmenden Datenverarbeiter zur Nutzung einer einheitlichen Technologieplattform verpflichten würden [258]. Dies würde manipulationssichere Hardware, Kontrolle der Dateneigner über die eingesetzte Software (und deren Integrität im Sinne von Nicht-Veränderbarkeit) und ein wirksames Digital Rights Management (DRM) voraussetzen. Dies sind Prämissen, die eine Umkehrung der vorliegenden Verteilung von Marktmacht erfordern würden. Es scheint heute wenig realistisch, dass Betroffene ihren Arbeitgebern, Krankenhäusern, Gewerkschaften oder gar staatlichen Institutionen die Verwendung einer von der Allgemeinheit kontrollierten DRM-Plattform vorschreiben könnten[259].

7.4.4 Weitergabe der Datenschlüssel

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Ein Verarbeiter könnte unvorsichtig mit dem Datenschlüssel umgehen, den er zur Inhaltsabfragen für die Daten eines Eigners erhalten hat, oder er gibt den Schlüssel absichtlich weiter, etwa gegen Geld. Der Erwerber des Schlüssels kann nun Inhaltsabfragen an den Aufbewahrer stellen.

Verteidigung

Gibt ein Verarbeiter den Datenschlüssel innerhalb der Empfängergruppe weiter, das heißt ist der Empfänger des Schlüssels ein beim Zertifizierer legitimiertes Gruppenmitglied, ist die Weitergabe ungefährlich. Der Empfänger der Weitergabe hätte ja jederzeit das Recht,

[258] vgl. [CrCP05] zu einer möglichen Architektur: Ein „Obligation Manager“ beim Verarbeiter würde die Nutzungsbeschränkungen prüfen und durch den „Enforcer“ umsetzen lassen. Auf Seiten des Betroffenen erhält der „Checker“ die Vollzugsmeldungen und gibt sie zur benutzerfreundlichen Auswertung an den „Trust Evaluator“ des Betroffenen weiter.

[259] Der umgekehrte Weg funktioniert seit geraumer Zeit. Man vergegenwärtige sich die DRM-Implementierung in Apples iTunes (vgl. [GrPu06]). Technisch bestehen also die Möglichkeiten.

genau diesen Schlüssel beim Zertifizierer abzufragen und würde ihn auch erhalten. Die Tatsache, dass über die Schlüsselabfrage im Falle einer bilateralen Weitergabe zwischen Empfängern kein Protokoll bei einer zentralen Stelle geschrieben wird, stellt kein Risiko dar. Für die juristische Absicherung der Datenübergaben sind ohnehin nur die Protokolle der Inhaltsabfragen (Ansprüche der Eigner gegen die Verarbeiter) bzw. Existenzanfragen (Ansprüche der Verarbeiter gegen die Dateneigner) relevant.

Ist hingegen der Empfänger des weitergegebenen Datenschlüssels kein Mitglied derselben Empfängergruppe wie der abgebende Verarbeiter, kann der Empfänger keinen Nutzen aus dem Schlüssel ziehen. Denn der Aufbewahrer schlägt immer die Gruppenmitgliedschaft eines Anfragenden in den veröffentlichten Listen des Zertifizierers nach, bevor er eine Inhaltsabfrage beantwortet. In dem hier genannten Fall würde er keinen Eintrag für den Anfragenden finden und somit würde dieser keine Antwort erhalten. Sollte der Abfragende ein Mitglied einer anderen Empfängergruppe sein, das bedeutet, der Aufbewahrer würde ihn in den Listen des Zertifizierers finden, erhielte er von diesem das entsprechende Datenpaket zur Antwort[260]. Der Eigner könnte in diesem Fall später nur über die Protokolle nachweisen, dass ein Missbrauch eines fremden Datenschlüssels durch einen Verarbeiter stattgefunden hat (dieser hat schließlich das Policy-Chiffrat, das er zur Abfrage genutzt hat, signiert). Bedroht man ein solches Verhalten mit Ausschluss aus dem System, wird zumindest der Preis des Missbrauchs hoch gesetzt. Man muss jedoch in die Betrachtung einbeziehen, dass es vermutlich für den Verarbeiter, der den Datenschlüssel weitergegeben hat, einfacher wäre, die gefragten Daten zu beschaffen und sie direkt statt des Schlüssels dem anderen Interessenten zu übergeben.

7.5 Angriffe von Außenstehenden

Standen die in den Kapiteln 7.1 bis 7.4 beschriebenen Angriffe unter der Annahme, dass die Kommunikation sicher ist, wird diese Annahme jetzt aufgelöst, um im Folgenden drei Angriffe auf Ebene der Kommunikationsschicht zu diskutieren.

7.5.1 Abhören von Kommunikation

Schutzziel

Vertraulichkeit

Motivation und Angriffsszenario

Mindestens so groß wie das Risiko, das der Missbrauch personenbezogener Daten durch bekannte Verarbeiter darstellt, ist das Missbrauchspotential durch unbekannte Dritte. Sie

[260] Die Forderung, der Aufbewahrer solle doch in der Policy nachschlagen, ob der anfragende Datenverarbeiter Mitglied der richtigen Empfängergruppe ist, geht fehl. Denn der Aufbewahrer kann die Policies nicht entschlüsseln.

könnten unerkannt die Datenflüsse verfolgen und ihre Zwecke aus den gewonnenen Erkenntnissen bedienen. Gelingt es ihnen, unerkannt zu bleiben und gehen sie umsichtig bei der weiteren Verwertung der ausgespähten Daten vor, ist für sie das Risiko, dass ihre Handlungen zu Sanktionen führen, niedriger als das der benannten Verarbeiter. Da zugleich durch den Verkauf von personenbezogenen Daten oder ganzen Identitäten Erlöse zu erzielen sind, besteht eine Motivation zu solchen Angriffen von außen.

Verteidigung

Die Vertraulichkeit der Datenübertragung ist im Verfahren durch Verschlüsselung aller Kommunikation, die über Netzwerke stattfindet, gesichert. Die zu Beginn beziehungsweise im Verlauf des Verfahrens zwischen allen miteinander direkt kommunizierenden Parteien ausgetauschten öffentlichen Schlüssel werden im Rahmen eines Public-Key-Verschlüsselungsverfahrens, etwa RSA, eingesetzt. Dieses wird im Rahmen einer Hybridanwendung dazu genutzt, symmetrische 3DES- oder AES-Sitzungsschlüssel zu chiffrieren und sicher an den jeweiligen Empfänger zu übertragen. Zu Beginn des Jahres 2012 gilt RSA unter Einsatz hinreichend langer Schlüssel (z.B. 3072 Bit)[261] und in Kombination mit zufällig gewählten symmetrischen Sitzungsschlüsseln in der veröffentlichten Forschung bei korrekter Implementierung als praktisch sicher.

Wie in Abschnitt 6.1.4 beschrieben, ist es von zentraler Bedeutung, dass auch die Daten, die bereits mit einem der One-Time-Pads chiffriert wurden, für ihre Übermittlung darüber hinaus mit dem Hybridverfahren verschlüsselt werden. Anderenfalls könnten Angreifer aus der Kombination mehrerer abgehörter Pakete den Schlüssel ermitteln.

Die wichtigste Variante eines Abhörangriffs auf das Verfahren ist ein Angriff mit bekanntem Klartext (know-plaintext Attacke). Schließlich ist die Menge der Kombinationen von Verarbeitungszweck und Datentyp endlich, ebenso lassen sich in eng umgrenzten Anwendungskontexten häufig verwendete Textblöcke ebenfalls raten. Daher muss der im Hybridverfahren verwendete symmetrische Verschlüsselungsalgorithmus ausreichend widerstandsfähig insbesondere gegenüber diesem Angriffstypus gewählt werden. Bouillaguet et al. zeigen, dass im Jahre 2011 etwa die besten bekannten Known-Plaintext Angriffe auf den Advanced Encryption Standard noch weit von praktischem Missbrauchspotential entfernt sind[262].

[261] vgl. die Empfehlungen des von der Europäischen Kommission geförderte European Network of Excellence in Cryptology II [Smar11], die jährlich über die Sicherheit von Krypto-Algorithmen und Schlüssellängen Bericht erstatten.

[262] vgl. [BDDK+11]. Die Autoren zeigen, dass die effizientesten bekannten Angriffe gegen einen mit drei Runden ausgeführten AES neun bekannte Klartexte, 2^{40} Verschlüsselungen und 2^{35} Bytes Speicherplatz benötigt. Für einen mit 6 Runden ausgeführten AES erhöht sich der Rechenbedarf auf 2^{120} Verschlüsselungen und es werden Chiffrate von $2^{108,5}$ bekannten Klartexten benötigt. Die in der Realität genutzten Implementierungen von AES verwenden jedoch, abhängig von der Schlüssellänge, 10 bis 14 Runden.

7.5.2 Man-in-the-Middle Angriffe gegen die Kommunikation

Schutzziele

Vertraulichkeit, Integrität, Verbindlichkeit

Motivation und Angriffsszenario

Bei einem Man-in-the-Middle Angriff täuscht ein Angreifer zwei oder mehr legitimen Teilnehmern eines Verfahrens vor, jeweils der von ihnen erwartete Kommunikationspartner zu sein. Kommunikationsflüsse, die er von einer Partei abfängt, liest und verändert er bei Bedarf und leitet sie anschließend an den beabsichtigten Empfänger weiter. Ebenso verfährt er mit den Rückmeldungen. Die anderen Kommunikationspartner bleiben also im Glauben, direkt miteinander zu kommunizieren. Ein erfolgreicher Man-in-the-Middle Angriff gegen verschlüsselte Kommunikation muss bereits initiiert werden, wenn die Kommunikationskanäle aufgebaut werden, das heißt bevor die legitimen Parteien Sitzungsschlüssel ausgetauscht haben. Hier kann der Angreifer noch seine eigenen Schlüssel in den Austausch einschleusen und sich damit Einblicke in den künftigen Nachrichtenaustausch sichern[263].

Verteidigung

Das vorgestellte Verfahren ist so anfällig gegen Man-in-the-Middle Angriffe wie jedes andere Verfahren, das verschlüsselte Kommunikation einsetzt und ist ebenso wie diese davor zu schützen: Die Integrität der öffentlichen Schlüssel muss gewahrt sein, wenn die beteiligten Parteien sie zu Beginn der Kommunikation beschaffen. Denn auf ihrer Basis wird alle weiterführende Kommunikation gesichert. Würden die öffentlichen Schlüssel der anderen Kommunikationspartner auf einem Schlüsselserver oder den Webseiten der Partner abgerufen, könnten Angreifer diese Anfragen abfangen und die zurückgemeldeten öffentlichen Schlüssel durch ihre eigenen ersetzen. Dies zu verhindern, sind Public-Key-Infrastrukturen (PKI) angetreten[264]. Sie formulieren verschiedene Modelle, durch die ein Kommunikationsteilnehmer sein Vertrauen festigen kann, den unveränderten öffentlichen Schlüssel des gewünschten Partners zu erhalten[265]. Oft sind sie in streng hierarchischen Modellen implementiert, bei denen eine zentrale, allgemein als vertrauenswürdig anerkannte Instanz die Authentizität der Schlüssel für andere Teilnehmer zertifiziert. Diese können die solchermaßen anerkannten Schlüssel wiederum nutzen, weitere Teilnehmer-Schlüssel zu zertifizieren uns so weiter. Problematisch ist dabei das hohe Risiko, das mit der Kompromittierung der obersten Instanz verbunden ist. An ihr hängt die Glaubwürdigkeit des Systems[266]. Fehlt eine übergeordnete Zentralinstanz, können

[263] Alternativ zur Manipulation am Datenverkehr gelangt ein Angreifer an Zertifikate, die es ihm erlauben, sich als eine andere Partei auszugeben, wie im Februar 2012 durch einen Kunden der Zertifizierungsinstanz Trustwave geschehen, vgl. http://www.heise.de/newsticker/meldung/Trustwave-verkaufte-Man-in-the-Middle-Zertifikat-1429722.html (Zugriff: 08.07.2012).

[264] vgl. u.a. [Schm09]

[265] vgl. [NDJB02]

[266] Die Kompromittierung der niederländischen Wurzel-Zertifizierungsinstanz DigiNotar führte beispielsweise im September 2011 dazu, dass der Angreifer Zertifikate für mehr als 500 namhafte Organisationen fälschen konnte, vgl. http://www.heise.de/newsticker/meldung/Aufsichtsbehoerde-untersagt-DigiNotar-das-Ausstellen-qualifizierter-Zertifikate-1344513.html (Zugriff: 12.05.2012).

verschiedene Vertrauensdomänen untereinander Zertifizierungen austauschen. Dies erhöht die Komplexität gegenüber dem streng hierarchischen Modell, da mit steigender Anzahl der Stationen, die zwischen einer zertifizierenden Instanz und dem Anfragenden der Authentizität eines Zertifikats liegt, gegebenenfalls die Glaubwürdigkeit und damit die Eignung des Zertifikats für sensible Einsatzgebiete sukzessive reduziert werden muss[267]. Einen anderen Ansatz nutzt das „Web of Trust". Hier kann jeder Teilnehmer die Schlüssel anderer Parteien zertifizieren. Jeder Teilnehmer kann selbst entscheiden, wie viele verschiedene Zertifikate ein öffentlicher Schlüssel besitzen muss und in welcher Beziehung die Zertifizierenden zu ihm selbst stehen müssen, so dass er der Authentizität des Schlüssels hinreichend vertraut[268].

7.5.3 Denial-of-Service Angriffe

Schutzziel

Verfügbarkeit

Motivation und Angriffsszenario

Mit einem Denial-of-Service (DoS) Angriff versuchen Angreifer, die Funktionsfähigkeit eines Systems in der Form zu beeinträchtigen, dass dessen legitime Nutzung verhindert oder über das tolerierbare Maß hinaus verzögert wird[269]. Physische Angriffe, wie das Zerstören von Hardware, sind mit Mitteln der physischen Sicherung zu verhindern. DoS im engeren Sinne bezieht sich aber auf entfernte Angriffe auf Server oder Dienste des Opfers, die sich der Eigenschaften verwendeter Netzwerkprotokolle, Betriebssysteme oder Applikationen bedienen. Sie nutzen Schwachstellen in Software und deren Konfiguration, um mit geringem Aufwand die Server des Anbieters zu überlasten. Bei verteilten DoS-Angriffen (Distributed DoS – DDoS) gelingt dies besonders gut, da eine Vielzahl von Rechnern, die zumeist unfreiwillig im Dienst des Angreifers stehen, koordiniert den Server des Opfers mit Anfragen überfrachtet. Opfer im vorgestellten Verfahren können insbesondere der Dienst des Aufbewahrers und die Dienste der Zertifizierungsinstanz sein. Die Motivation der Angreifer kann einem breiten Spektrum entstammen. Carl Denis nennt die Kategorien „Cyber-Warfare" (Durchsetzung weltanschaulicher, wirtschaftlicher oder militärischer Interessen), „Organisierte Kriminalität" (insbesondere Erpressung des Dienstanbieters oder Verbessern der Marktposition eines Wettbewerbers) und „Die kleine Rache" (beispielsweise durch enttäuschte ehemalige Mitarbeiter oder in Rechtsstreitigkeiten Unterlegene)[270].

Verteidigung

Vollständig zu vermeiden sind DoS-Angriffe nicht. Insbesondere potentielle Opfer eines Angriffs können kaum verhindern, dass ein solcher begonnen wird, da sie keine Kontrolle

[267] vgl. [KaTD11]
[268] vgl. [Garf94]
[269] vgl. [CLFF+00]
[270] vgl. [Deni09]

über die angreifende Infrastruktur haben. Sie können aber die schadhaften Auswirkungen begrenzen. Dies beginnt bei der Auswahl geeigneten Serverbetreiber und Netzdiensteanbieter. Wie stark ein DoS-Angriff den eigenen Dienst beschädigt, hängt davon ab, ob die Betreiber bereits ein zuverlässiges Filtern eingehenden Netzwerkverkehrs durchführen und dabei Anfragen frühzeitig unterbinden, die außerhalb der normalen Parameter liegen. Weiterhin können durch DoS-Angriffe erzeugt Anfragespitzen durch geeignetes Load Balancing, Proxies und Reserve-Server entschärft werden[271]. Zusätzlich sind Schwachstellen in der Server-Software regelmäßig zu schließen, die von den höher entwickelten DoS-Angriffsszenarien ausgenutzt werden können, indem eine einfache, fehlerhaft gestellte Anfrage im angegriffenen Dienst Schleifen, Wartezeiten oder umfangreiche Fehlermeldungen erzeugt.

[271] vgl. [Hatt09]

8 Bewertung und Zusammenfassung

8.1 Auswertung des PDG/v

Abbildung 29 zeigt den Steckbrief des Verfahrens PDG/v. Wie gefordert erfolgt die Formulierung der Policies beim Betroffenen. Dementsprechend bilden sie auch seine persönlichen Datenschutz-Präferenzen ab. Sie basieren auf einem Vokabular, das von allen Parteien vorab zu vereinbaren ist. Die Datenschutzregeln in PDG/v sind Sticky Policies und ihre Durchsetzung im System nicht durch andere Parteien als den Betroffenen umgehbar, bis sie gegenüber einem Verarbeiter aufgedeckt wurden. Bis zu diesem Punkt ist die Policy-Durchsetzung also präventiv, es wird der vollständige Lebenszyklus der Daten abgedeckt. Nach der erfolgreichen Inhaltsabfrage durch einen Verarbeiter stellt die Protokollierung aller Abfragen die Grundlage für detektivische Kontrollmaßnahmen zur Verfügung.

PDG/v

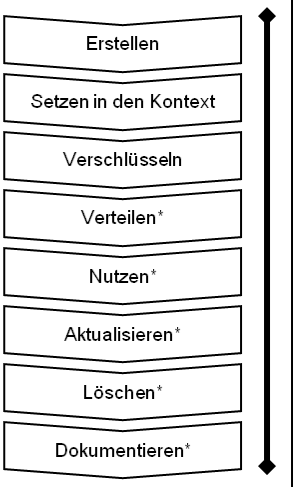
Policy-Management

Formulierungsort der Policies	Beim Betroffenen
Perspektive der Policy-Formulierung	Datenschutz-Präferenzen des Dateneigners
Management von Änderungen	Migration auf die neue Policy
Sprachen zur Formulierung	Alle Betroffenen-spezifischen oder proprietär
Bindung an die Daten	Technisch nicht umgehbar (bis Datenaufdeckung)

Durchsetzung der Policies

Zeitpunkt / Ort	Beim Betroffenen	Beim Verarbeiter	Bei Dritten
Präventiv, vor Übermittlung			●
Präventiv, vor Verarbeitung			
Detektivisch, nach Verarbeitung			
Nur Willenserklärung			

Wirkung im Daten-lebenszyklus



*Nur vor der Inhaltsabfrage

Abbildung 29: Steckbrief PDG/v

8.2 Zusammenfassung

Die Ergebnisse der vorliegenden Arbeit werden nun in Beantwortung der eingangs formulierten Forschungsfragen zusammengefasst:

1. Welche Optionen für die technische Durchsetzung von Datenschutzregeln bestehen und wie kann man sie systematisieren?

   Zunächst wurde gezeigt, dass Verarbeitungskontrolle als wirksames Mittel zum Selbstdatenschutz auf den Bereich der aktiven und rational bewussten Datenbereitstellung durch Betroffene beschränkt ist. Hier bewirkt sie im Einsatz als datenschutzfreundliche Technik eine Reduktion des Missbrauchspotentials und erhöht die Transparenz der Datenverarbeitung.

   Es lassen sich Kriterien definieren, die eine Systematisierung der Verfahren erlauben. Die hier verwendeten Kriterien sind im Policy-Management der Formulierungsort der Policies, die Perspektive der Policy-Formulierung, das Management von Änderungen, Sprachen zur Formulierung und die Art der Policy-Bindung an die Daten. Bei der Durchsetzung von Policies dienen zum einen der Ort und zum anderen der Zeitpunkt der Durchsetzung als Kriterien. Schließlich unterscheiden sich die Verfahren im Umfang ihrer Wirkung im Lebenszyklus der zu schützenden Daten.

2. Welche Aussagen kann man daraus über die Qualität entsprechender Verfahren im Sinne des Datenschutzes treffen?

   Der Schutzbedarf des Einzelnen ist subjektiv, so dass eine vollständige qualitative Bewertung von Verfahren der Verarbeitungskontrolle nicht in allgemein gültigen Aussagen möglich ist. Auch wenn die Gewichtung der Kriterien gegeneinander nur im konkreten Einsatzkontext und unter Kenntnis der Risikoaffinitäten aller Beteiligter möglich ist, so erlaubt ein Vergleich der Verfahren innerhalb der Kriterien durchaus eine qualitative Messung. So ist etwa ein Verfahren einem anderen hinsichtlich des Kriteriums „Wirkung im Datenlebenszyklus“ überlegen, wenn es alle Lebenszyklus-Phasen abdeckt, auf die sich auch das zweite erstreckt, und mindestens eine weitere Phase einbezieht. Ebenso sind präventiv wirkende Verfahren den rein detektivisch wirksamen vorzuziehen. Sticky Policies sind ein weiteres Qualitätsmerkmal, durch das sich fortgeschrittene Verfahren auszeichnen.

3. Wie sind bestehende Verfahren der Praxis und theoretische Ansätze in die Systematik einzuordnen? Wie werden sie bewertet?

   Als vordergründig auf die Bedürfnisse des Betroffenen ausgerichtetes theoretisches Verfahren wurde die exklusive Speicherung beim Betroffenen diskutiert. Sie eignet sich zur Betrachtung der optimierten Annahmen, scheitert aber an der Realisierbarkeit. Zum einen bestehen keine Möglichkeiten, Verarbeiter vollständig daran zu hindern, Daten zu speichern, zum anderen hat sich gezeigt, dass dies im

Sinne der Effizienz eines Verfahrens nicht wünschenswert wäre, ja teilweise sogar den Interessen des Betroffenen zuwiderliefe.

Die Untersuchung von Trusted Computing und den Techniken des Digital Rights Management zeigte, dass solche Verfahren das Potential aufweisen, im Sinne des Datenschutzes eingesetzt zu werden. Jedoch spricht die aktuelle Machtverteilung zwischen Betroffenen und Verarbeitern gegen den flächendeckenden Einsatz. Würde dieser auf regulativem Wege erzwungen, bestünde dennoch die Schwierigkeit, die verschiedenen Interessenlagen der Betroffenen auf einen Nenner zu bringen.

So wurden stellvertretend für Verfahren im praktischen Einsatz die klassische Access Control, Hippokratische Datenbanken und Privacy Management Systeme evaluiert. Es zeigte sich, dass die Abdeckung des Datenlebenszyklus mit fortschreitender Evolution der Verfahren zugenommen hat. Zudem hat sich das Paradigma der Sticky Policies verfestigt und wird zu Datenlizenzen ausgebaut. Policies reflektieren in steigendem Maße die Präferenzen der Betroffenen.

4. Welche Entwicklungsziele sollten angegangen werden, um ein höheres Datenschutzniveau zu erreichen?

   Den verbreiteten Verfahren ist sämtlich die „Administratoren-Lücke“ zu Eigen, und sie beachten nicht die Daten-Schutzwürdigkeit der Policies. Ebenso begleiten die Verfahren den Betroffenen nicht durch den Prozess der Datenbereitstellung. Dies sollten also die vordringlichen Entwicklungsziele sein, wenn ein höheres Datenschutzniveau erreicht werden soll.

   Um die Administratoren-Lücke zu schließen, muss der kryptographische Schutz der personenbezogenen Daten bereits vollständig aufgebaut sein, bevor die Daten den Einflussbereich des Betroffenen verlassen. Dies bedingt den Einsatz entsprechender Module und Verfahrensschritte in der Verantwortung des Betroffenen, was gleichzeitig Möglichkeiten zur Verschleierung der Policies vor den Augen Dritter eröffnet.

5. Welche Anforderungen soll ein Verfahren erfüllen, das im Bereich der bewussten Datenbereitstellung für situativ bedingte Abrufe dieses Niveau anstrebt?

   Grundsätzlich muss ein Verfahren den allgemeinen Anforderungen an datenschützerisches Handeln folgen. Die Vertraulichkeit, Verfügbarkeit und Integrität der Daten muss im Prozess gewahrt werden. Die Verbindlichkeit ist durch entsprechende Nachweismöglichkeiten der relevanten Aktivitäten bei Betroffenen und Verarbeitern zu sichern.

   Darüber hinaus wurde erarbeitet, dass die Formulierung der Policies möglichst im Einflussbereich des Betroffenen liegen soll. Die gewählten Anwendungsszenarien der Daten-Kategorie „Applikation“ erfordern zudem, Policies asynchron durchzusetzen, also ohne dass Betroffener und Verarbeiter zum gleichen Zeitpunkt

aktiv werden müssen. Zudem sind variable Empfängergruppen zu realisieren, deren individuelle Mitglieder zum Zeitpunkt der Policy-Formulierung noch nicht feststehen. Das gesuchte Verfahren soll personenbezogene Daten bis zum Zeitpunkt der konkret veranlassten Nutzung geheim halten, zuverlässig liefern und die Präferenzen des Betroffenen dynamisch implementieren.

Diese Anforderungen wurden in Kapitel 4.2 konkretisiert.

6. Wie ist ein solches Verfahren zu gestalten? Welche Protokolle, Algorithmen und Systemkomponenten eignen sich zum Einsatz?

   Das vorgestellte Verfahren PDG/v benötigt neben den Betroffenen und den Datenverarbeitern eine Aufbewahrungsinstanz, bei der Schlüssel- und Datenpakete hinterlegt sind, die selbst aber keinen Zugriff auf die Daten erlangen kann. Zur Abbildung der variablen Empfängergruppen wurde eine Zertifizierungsinstanz eingeführt, die zusätzlich eine Rolle im Schlüsselaustausch wahrnimmt. Die Parteien und Protokollschritte wurden ausführlich in Kapitel 5 beschrieben. Es folgte eine Diskussion der kryptographischen Optionen zum Schlüsselaustausch. Aus der Notwendigkeit, einen Algorithmus mit kommutativer Schlüsselanwendung zu verwenden, fiel die Wahl auf das One-Time-Pad als Kern des kryptographischen Protokolls.

   Weiterhin wurden in Abschnitt 6 generisch die Komponenten einer möglichen Applikationsarchitektur für das Verfahren erarbeitet. Die Implementierung des Verfahrens an der Universität Hamburg steht zudem als Proof-of-Concept zur Verfügung.

7. Wie verhält sich das entwickelte Verfahren gegenüber Angriffen von Teilnehmern und von Außenstehenden?

   Die Diskussion der Angreifermodelle zeigte, dass PDG/v die formulierten Anforderungen erfüllt und bestehende Schwachstellen durch zusätzliche Kontrollmaßnahmen weitgehend beseitigt werden können. Eine generelle Anfälligkeit gegenüber Denial-of-Service-Attacken teilt das Verfahren jedoch mit vielen anderen Systemen.

   Die Anforderungen an die Dienstleister sind relativ gering. Gegen die Neugierde von Treuhänder und Zertifizierer ist das System gewappnet. Beide müssen jeweils ein grundsätzliches Interesse am Funktionieren des Systems haben, anderenfalls könnten sie es durch Verweigerung der Datenübermittlung sabotieren. Kompromittieren kann es jedoch keiner der beiden aus eigenen Stücken. Für das Gewinnen von Einsicht in die Datensätze sowie eine konstruktive Fälschung von Protokollen oder Inhalten wäre in jedem Fall die Kooperation mit mindestens einer weiteren der beteiligten Parteien notwendig.

   Das Hauptrisiko besteht in einer böswilligen Kooperation von Aufbewahrer und Zertifizierer. Würden sie die bei ihnen hinterlegten One-Time-Pads entgegen den

Vorgaben des Protokolls miteinander kombinieren, würden sie PDG/v kompromittieren. Eine Option zur weitgehenden Linderung dieses Risikos wurde erarbeitet: Durch Aufteilen der Zertifizierungsinstanz auf mehrere Parteien kann die Zahl der zur Kompromittierung notwendigen Kooperationspartner und damit der Schwierigkeitsgrad des Datenmissbrauchs erhöht werden.

8. Erreicht das neue Verfahren die angestrebte Verbesserung im Datenschutz-Niveau? Welcher weitere Entwicklungsbedarf besteht?

   Mit dem vorgestellten Protokoll gelingt es dem Dateneigner, eine anonyme Datenablage mit „Sticky Policies“ vorzunehmen. Deren Zeitpunkt wählt er selbst, um die konkrete Datenherausgabe an einen legitimierten Verarbeiter zum richtigen Anlass muss er sich nicht kümmern. Die Verwaltung der Gruppenmitgliedschaften obliegt einem Zertifizierer, der zusammen mit dem Treuhänder-Service auch den Schlüsselaustausch betreut. Das Protokoll gewährt den beiden Dienstleistern nur wenig Einblick in die Kommunikation – es werden nur die Abfragen der Verarbeiter wahrgenommen – über Dateneigner, den Inhalt der Policies oder der Datensätze erfahren sie nichts. Die Abfragen und Datenübermittlungen sind vollständig protokolliert und nicht abzustreiten.

   Der Schutz des Eigners vor Missbrauch seiner Daten steht im Vordergrund des Systems. Daneben ist ein zu untersuchendes Feld, wie auch die Interessen der Verarbeiter verstärkt durchgesetzt werden können. Insbesondere beruht der Nachweis aus einer Existenzabfrage auf der Beobachtung, ob zu einer bestimmten Policy ein Datensatz abgelegt wurde. Eine semantische Prüfung, ob der abgelegte Inhalt auch den Anforderungen der Policy genügt oder gar, ob er den wahren Verhältnissen des Eigners entspricht, findet aktuell nicht statt. Die Einführung einer zusätzlichen Instanz[272], die vor Verschlüsselung durch den Eigner die Daten einer Plausibilisierung unterzieht und nach Qualitätskriterien beglaubigt, ist ein möglicher Ansatz hierfür.

   Die hinterlegten personenbezogenen Daten bedürfen nicht nur hin und wieder einer Aktualisierung durch den Dateneigner. Die Speicherung beim Aufbewahrer hat gegebenenfalls auch weiteren Verpflichtungen nachzukommen, etwa der Löschung der Daten nach einem bestimmten Zeitraum ohne Zutun des Eigners, regelmäßiger Kontrolle der Integrität oder Verpflichtungen zur Auskunft über gespeicherte Daten. Datenschutzkonformes Management von personenbezogenen Daten in Unternehmenssystemen wird nicht alleine durch entsprechende Zugangs- und Verarbeitungskontrollen sichergestellt, sondern muss durch die Bindung von Verpflichtungen an die Daten ergänzt werden[273]. Der Lebenszyklus der Speicherung personenbezogener Daten soll so vollständig abgedeckt werden[274].

[272] Etwa in der Rolle eines „Auditors“, vgl. [ShSB08].
[273] vgl. [Casa05]
[274] vgl. [CaBe07]

Der Dateneigner kann im Protokoll gegenüber Treuhänder und Zertifizierer anonym auftreten, während die Beobachtbarkeit und entsprechende Kontrollmöglichkeit der anderen Parteien im Sinne des Verfahrens ist. Benötigen die Verarbeiter im Anwendungsszenario einen starken Nachweis der Identität der Dateneigner, müsste deren Signierfunktion beispielsweise als fortgeschrittene oder qualifizierte elektronische Signatur[275] implementiert werden.

Die Einsatzmöglichkeiten des Protokolls sind über den Schutz rein personenbezogener Daten hinaus erweiterbar. Dazu sind lediglich die Definitionen der Datensatztypen und Verarbeitungszwecke dem gewählten Szenario anzupassen, sowie die Anforderungen an eine Zertifizierung als Mitglied der Verarbeitergruppen zu spezifizieren.

Bezieht man alle Arten von Daten ein, deren Aufdeckung gegenüber bestimmten Parteien für definierte Einsatzzwecke zu reservieren ist, findet sich etwa im Management von Vertragsbeziehungen ein weites Feld: Ein Lieferant verpflichtet sich vertraglich, für seinen Kunden ein Produkt mit spezifischen Eigenschaften herzustellen, und diese Eigenschaften beruhen auf der Anwendung von Verfahren oder Rezepturen, die als Unternehmensgeheimnisse nicht preisgegeben werden sollen. Dann kann man sich darauf einigen, die fraglichen Informationen gemäß dem geschilderten Protokoll bei einer Treuhänderinstanz zu hinterlegen, und die legitime Abfrage der Daten durch den Kunden auf die Zwecke zu beschränken, die sich aus einer möglichen Verletzung von Richtlinien zum Umweltschutz oder Patentansprüchen Dritter ergeben. So kann sich der Kunde gegenüber rechtlichen Risiken absichern, ohne dass der Lieferant seine Geschäftsgeheimnisse unnötig aufdeckt. Ebenso können Details von Preiskalkulationen, interne Bewertungen von Finanzprodukten oder Nachweise im Sinne des Geldwäschegesetzes die zu schützenden Daten darstellen[276]. Für Privatpersonen ließe sich auch die Weitergabe von Erfindungen, Ideen oder literarischen Leistungen an Interessierte (wie Verlage oder Unternehmen) abbilden, ohne die einzelnen Mitglieder der Zielgruppe exakt zu kennen, wobei gleichzeitig alle stattgefundenen Abrufe zuverlässig nachgewiesen werden können.

Zur Optimierung des Verfahrens PDG/v kann an verschiedenen Aspekten weitere Forschung betrieben werden. Insbesondere scheint die Untersuchung sinnvoll, wie weit das Risiko einer Kooperation von Aufbewahrer und Zertifizierer in betrügerischer Absicht über die vorgeschlagene Aufteilung der Zertifizierer-Rolle hinaus reduziert werden kann. Praktische Aspekte des Verfahrens, wie Handhabbarkeit und Performanz, sind anhand der prototypischen Implementierung an der Universität Hamburg zu prüfen. Künftige Entwicklungen im Bereich des Digital Rights Management mögen zudem Forschungsansätze für durchsetzungsstarke Verarbeitungskontrolle auch nach der Aufdeckung der Daten bieten.

[275] Im Sinne des Bundesgesetzes über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz).
[276] vgl. [Wagn11]

Literaturverzeichnis

[AABG+05] *Agrawal*, Rakesh; *Asonov*, Dimitri; *Bayardo*, Roberto; *Grandison*, Tyrone; *Johnson*, Christopher: Managing Disclosure of Private Financial Data with Hippocratic Databases 2005, http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/nc_hdb_white_paper_finance.pdf (Zugriff: 02.01.2012)

[AABG+05a] *Agrawal*, Rakesh; *Asonov*, Dimitri; *Bayardo*, Roberto; *Grandison*, Tyrone; *Johnson*, Christopher; *Kiernan*, Jerry: Managing Disclosure of Private Health Data with Hippocratic Databases 2005, http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/nc_hdb_white_paper_health.pdf (Zugriff: 02.01.2012)

[Abel06] *Abel*, John: Oracle E-Business Suite Security, McGraw-Hill/Osborne, New York 2006

[ABFK+04] *Agrawal*, Rakesh; *Bayardo*, Roberto; *Faloutsos*, Christos; *Kiernan*, Jerry; *Rantzau*, Ralf; *Srikant*, Ramakrishnan: Auditing Compliance with a Hippocratic Database; Proc. of the 30th Int'l Conf. on Very Large Databases (VLDB 2004), Toronto, Canada 2004

[ABGG+02] *Alamäki*, Tero; *Björksten*, Margareta; *Dornbach*, Péter; *Gripenberg*, Casper; *Györbíró*, Norbert; *Márton*, Gábor; *Németh*, Zoltán; *Skyttä*, Timo; *Tarkiainen*, Mikko: Privacy Enhancing Service Architectures; in: *Dingledine*, Roger; *Syverson*, Paul (Hrsg.): Privacy Enhancing Technologies, Second International Workshop, PET 2002 San Francisco, CA, USA, Revised Papers; Lecture Notes in Computer Science Band 2482, S.99-109, Springer-Verlag, Berlin 2002

[AgAS04] *Agrawal*, Rakesh; *Asonov*, Dimitri; *Srikant*, Ramakrishnan: Enabling Sovereign Information Sharing Using Web Services; in: *Weikum*, Gerhard; *König*, Arnd Christian; *Dessloch*, Stefan (Hrsg.): Proc. of the ACM SIGMOD Conference on Management of Data, Paris, S.873-877, ACM Press, New York 2004

[AHKP+03] *Ashley*, Paul; *Hada*, Satoshi; *Karjoth*, Günter; *Powers*, Calvin; *Schunter*, Matthias: Enterprise Privacy Authorization Language (EPAL 1.2), W3C Member Submission 2003, http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/ (Zugriff: 03.06.2012)

[AHKS02] *Ashley*, Paul; *Hada*, Satoshi; *Karjoth*, Günter; *Schunter*, Matthias: E-P3P Privacy Policies and Privacy Authorization; in: WPES '02 Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society, New York, S.103-109, ACM Press, New York 2002

[AKSX02] *Agrawal*, Rakesh; *Kiernan*, Jerry; *Srikant*, Ramakrishnan; *Xu*, Yirong: Hippocratic Databases; Proc. of the 28th Int'l Conf. on Very Large Databases (VLDB 2002), S.143-154, Hong Kong, China 2002

[AKSX04] *Agrawal*, Rakesh; *Kiernan*, Jerry; *Srikant*, Ramakrishnan; *Xu*, Yirong: Order preserving encryption for numeric data; Proceedings of the 2004 ACM SIGMOD

international conference on Management of data, S.563-574, ACM Press, New York 2004

[AKSX05] *Agrawal*, Rakesh; *Kiernan*, Jerry; *Srikant*, Ramakrishnan; *Xu*, Yirong: XPref: a preference language for P3P; in: Computer Networks: The International Journal of Computer and Telecommunications Networking - Web security, Volume 48 Issue 5, S.809-827, Elsevier North-Holland, New York 2005

[Alma04] *IBM Almaden Research Center*: Intelligent Information Systems / Hippocratic Database 2004, www.almaden.ibm.com/software/disciplines/iis/ bzw. (Zugriff: 17.03.2012)

[Ande03] *Anderson*, Ross: 'Trusted Computing' Frequently Asked Questions, - TC / TCG / LaGrande / NGSCB / Longhorn / Palladium / TCPA, Version 1.1 2003, http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html (Zugriff: 12.03.2012)

[Ande04] *Anderson*, Anne: An Introduction to the Web Services Policy Language (WSPL); in: 5th IEEE International Workshop on Policies for Distributed Systems and Networks, Yorktown Heights, New York, 7-9 June 2004, IEEE Computer Society Press, Washington 2004

[Ande05] *Anderson*, Anne: A Comparison of Two Privacy Policy Languages: EPAL and XACML 2005, http://labs.oracle.com/techrep/2005/smli_tr-2005-147.pdf (Zugriff: 17.03.2012)

[Ande06] *Anderson*, Anne: Sun Position Paper: W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement 2006, http://www.w3.org/2006/07/privacy-ws/papers/17-anderson-position/ (Zugriff: 17.03.2012)

[ANSI04] *American National Standards Institute*; *INCITS*: 359-2004: Role Based Access Control 2004, http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+INCITS+359-2004&source=google&adgroup=incits&keyword=ANSI%20incits%20359%202004&gclid=CJbXqOCjpa0CFcYl3godUzeTzg (Zugriff: 17.03.2012)

[AsFr02] *Asonov*, Dmitri; *Freytag*, Johann-Christoph: Almost Optimal Private Information Retrieval; in: *Dingledine*, Roger; *Syverson*, Paul (Hrsg.): Privacy Enhancing Technologies, Second International Workshop, PET 2002 San Francisco, CA, USA, Revised Papers; Lecture Notes in Computer Science Band 2482, S.209-223, Springer-Verlag, Berlin 2002

[AsPS02] *Ashley*, Paul; *Powers*, Calvin; *Schunter*, Matthias: From Privacy Promises to Privacy Management, A New Approach for Enforcing Privacy Throughout an Enterprise; in: *Hempelmann*, Christian F.; *Raskin*, Victor (Hrsg.): Proceedings - New Security Paradigms Workshop 2002, September 23-26, Virginia Beach, VA, USA, S.43-50, ACM Press, New York 2002

[BaDS04] *Backes*, Michael; *Dürmuth*, Markus; *Steinwandt*, Rainer: An Algebra for Composing Enterprise Privacy Policies; in: *Samarati*, Pierangela; *Ryan*, Peter; *Gollmann*, Dieter; *Molva*, Refik (Hrsg.): Computer Security - ESORICS 2004, 9th European Symposium on Research in Computer Security, Sophia Antipolis, France,

Proceedings; Lecture Notes in Computer Science Band 3193, S.33-52, Springer-Verlag, Berlin 2004

[Baer02a] *Baeriswyl*, Bruno: Vom eindimensionalen zum mehrdimensionalen Datenschutz - Tendenzen der Rechtsentwicklung; in: *Baeriswyl*, Bruno; *Rudin*, Beat (Hrsg.): Perspektive Datenschutz, Praxis und Entwicklungen in Recht und Technik, S.47-65, Schulthess Juristische Medien, Zürich 2002

[Baha10] *Bahadur*, Gary: What Is Data Lifecycle Management?, http://ezinearticles.com/?What-Is-Data-Lifecycle-Management?&id=5056508 (Zugriff: 17.03.2012)

[BaMR04] *Barth*, Adam; *Mitchell*, John; *Rosenstein*, Justin: Conflict and Combination in Privacy Policy Languages; in: *Atluri*, Vijay; *Syverson*, Paul F., *De Capitani di Vimercati*, Sabrina (Hrsg.): WPES '04 : proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, Washington DC, USA, S.45-46, ACM Press, New York 2004

[BaSr03] *Bayardo*, Robert J.; *Srikant*, Ramakrishnan: Technological Solutions for Protecting Privacy; in: IEEE Computer, Volume: 36 Issue 9, S.115-118, 2003

[Bäum02] *Bäumler*, Helmut: Datenvermeidung und Datensparsamkeit; in: *Baeriswyl*, Bruno; *Rudin*, Beat (Hrsg.): Perspektive Datenschutz, Praxis und Entwicklungen in Recht und Technik, S.351-368, Schulthess Juristische Medien, Zürich 2002

[BBDG+11] *Backes*, Julian; *Backes*, Michael; *Dürmuth*, Markus; *Gerling*, Sebastian; *Lorenz*, Stefan: X-pire! A digital expiration date for images in social networks; arXiv:1112.2649v1 [cs.CR] 12 Dec 2011, http://arxiv.org/abs/1112.2649 (Zugriff: 17.03.2012)

[BCSS+06] *Bilger*, Mike; *O'Conner*, Luke; *Schunter*, Matthias; *Swimmer*, Morton; *Zunic*, Nev: Data-centric Security, Enabling business objectives to drive security, 2006, http://www-05.ibm.com/services/fi/cio/risk/gov_wp_data_centric.pdf (Zugriff: 17.03.2012)

[BDDK+11] *Bouillaguet*, Charles; *Derbez*, Patrick; *Dunkelman*, Orr; *Keller*, Nathan; *Fouque*, Pierre-Alain: Low Data Complexity Attacks on AES, Paris 2011, http://www.di.ens.fr/~fouque/pub/tissec11.pdf (Zugriff: 21.02.2012)

[BDOS08] *Bonatti*, Piero; *De Coi*, Juri Luca; *Olmedilla*, Daniel; *Sauro*, Luigi: Protune: A framework for semantic web policies; in: Proceedings of The 7th International Semantic Web Conference (ISWC 2008), Karlsruhe, Germany, Lecture notes in computer science Band 5318, Springer-Verlag, Berlin 2008

[Beal10] *Beales*, Howard: The Value of Behavioral Targeting, 2010, http://www.networkadvertising.org/pdfs/Beales_NAI_Study.pdf (Zugriff: 17.03.2012)

[Bech04] *Bechtold*, Stefan: Digital Rights Management nach der Urheberrechtsnovelle; in: *Büllesbach*, Alfred; *Dreier*, Thomas (Hrsg.): Wem gehört die Information im 21. Jahrhundert? Proprietäre versus nicht proprietäre Verwertung digitaler Inhalte;

Informationstechnik und Recht Band 13, S.145-161, Verlag Dr. Otto Schmidt, Köln 2004

[BeGü04] *Becker*, Eberhard; *Günnewig*, Dirk: Digital Rights Management & Trusted Computing - Technische Aspekte; in: *Büllesbach*, Alfred; *Dreier*, Thomas (Hrsg.): Wem gehört die Information im 21. Jahrhundert? Proprietäre versus nicht proprietäre Verwertung digitaler Inhalte; Informationstechnik und Recht Band 13, S.11-36, Verlag Dr. Otto Schmidt, Köln 2004

[BeKö01] *Berthold*, Oliver; *Köhntopp*, Marit: Identity Management Based on P3P; in: *Federrath*, Hannes (Hrsg.): Designing privacy enhancing technologies, Design Issues in Anonymity and Unobservability; Lecture notes in computer science Band 2009, S.141-160, Springer-Verlag, Berlin 2001

[BeMW03] *Bellare*, Mihir; *Micciancio*, Daniele; *Warinschi*, Bogdan: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions; in: *Biham*, Eli (Hrsg.): Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Application of Cryptographic Techniques, Warsaw, Poland, Proceedings; Lecture Notes in Computer Science Band 2656, S.614-629, Springer-Verlag, Berlin 2003

[Bena05] *Benantar*, Messaoud (Hrsg.): Access Control Systems, Security, Identity Management and Trust Models, Springer-Verlag, Berlin 2005

[BeSW04] *Beutelspacher*, Albrecht; *Schwenk*, Jörg; *Wolfenstetter*, Klaus-Dieter: Moderne Verfahren der Kryptographie, Von RSA zu Zero-Knowledge, 5. Auflage, Vieweg, Wiesbaden 2004

[BHHJ+12] *Baumann*, Christian; *Hodjov*, Ahmed; *Hoepfner*, Stephan; *Jerger*, Roman; *Keitzel*, Dennis; *Kuhlmann*, Hannes; *Lauterbach*, Stephan: Digitaler Umschlag, Dokumentation zum SVS-Masterprojekt, Betreuer: *Federrath*, Hannes; Universität Hamburg, Fakultät für Mathematik, Informatik und Naturwissenschaften, 2012

[BiGr06] *Bizer*, Johann; *Grimm*, Rüdiger: Privacy4DRM: Innovationen für den Kunden - nicht gegen ihn!; in: DuD 30 (Feb. 2006) S.66, Vieweg, Wiesbaden 2006

[BiGW06] *Bizer*, Johann; *Grimm*, Rüdiger; *Will*, Andreas: Privacy4DRM: Nutzer- und datenschutzfreundliches Digital Rights Management; in: DuD 30 (Feb. 2006) S.69-73, Vieweg, Wiesbaden 2006

[Bitz05] *Bitz*, Gunter: Informationsschutz im Unternehmen, Policy Enforcement mit Trusted Computing; in: DuD 29 (Sep. 2005) S.531-536, Vieweg, Wiesbaden 2005

[Bize02] *Bizer*, Johann: Datenspeicherung in zentralen und peripheren Netzen versus SmartCards - wozu digitale Signaturen in der öffentlichen Verwaltung?; in: *Möller*, Klaus Peter; *von Zezschwitz*, Friedrich (Hrsg.): Verwaltung im Zeitalter des Internet, Vernetzte Verwaltung und Datenschutz; Forum Datenschutz Band 9, S.19-58, Nomos-Verlag, Baden-Baden 2002

[Bize04] *Bizer*, Johann: Strukturplan modernes Datenschutzrecht; in: DuD 28 (Jan. 2004) S.6-14, Vieweg, Wiesbaden 2004

[BlBO03] *van Blarkom*, G.W.; *Borking*, J.J.; *Olk*, J.G.E.: Handbook of Privacy and Privacy-Enhancing Technologies, The case of Intelligent Software Agents, 2003, www.andrewpatrick.ca/pisa/handbook/Handbook_Privacy_and_PET_final.pdf (Zugriff: 17.03.2012)

[BoFr01] *Boneh*, Dan; *Franklin*, Matthew K.: Identity-Based Encryption from the Weil Pairing; in: Advances in Cryptology - Crypto2001, Lecture Notes in Computer Science Band 2139, S.213-229, Springer-Verlag, Berlin 2001

[BoOl05] *Bonatti*, Piero; *Olmedilla*, Daniel: Driving and monitoring provisional trust negotiation with metapolicies; In: 6th IEEE Policies for Distributed Systems and Networks (POLICY 2005), S.14-23, Stockholm, Sweden, IEEE Computer Society Press, Washington 2005

[Bran00] *Brands*, Stefan A.: Rethinking Public Key Infrastructures and Digital Certificates, Building in Privacy, MIT Press, Cambridge, London 2000

[Bran05] *Brandl*, Hans: Trusted Computing - Aktuelle Anwendungen, Welche Funktionen sind heute schon nutzbar?; in: DuD 29 (Sep. 2005) S.537-541, Vieweg, Wiesbaden 2005

[BrCC04] *Brickell*, Ernie; *Camenisch*, Jan; *Chen*, Liqun: Direct Anonymous Attestation; in: *Pfitzmann*, Birgit; *Liu*, Peng (Hrsg.): Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington DC, USA, S.132-145, ACM Press, New York 2004

[BrHS04] *Bradshaw*, Robert W.; *Holt*, Jason E.; *Seamons*, Kent E.: Concealing Complex Policies with Hidden Credentials; in: *Pfitzmann*, Birgit; *Liu*, Peng (Hrsg.): Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington DC, USA, S.146-157, ACM Press, New York 2004

[BrRo04] *Brandl*, Hans; *Rosteck*, Thomas: Technik, Implementierung und Anwendung des Trusted Computing Group-Standards (TCG), Sichere Plattformen ermöglichen neue Sicherheitsniveaus; in: DuD 28 (Sep. 2004) S.529-538, Vieweg, Wiesbaden 2004

[BSI11] *Bundesamt für Sicherheit in der Informationstechnik* (Hrsg.): Mindestanforderungen zur Informationssicherheit bei eCommerce-Anbietern, Version 1.2 2011, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindestanforderungen/Mindestanforderungen-eCommerce-Anbieter.pdf?__blob=publicationFile (Zugriff: 17.03.2012)

[Büll04] *Büllesbach*, Alfred: Datenschutzrechtliche Aspekte des Digital Rights Management; in: *Büllesbach*, Alfred; *Dreier*, Thomas (Hrsg.): Wem gehört die Information im 21. Jahrhundert? Proprietäre versus nicht proprietäre Verwertung digitaler Inhalte; Informationstechnik und Recht Band 13, S.163-175, Verlag Dr. Otto Schmidt, Köln 2004

[CaBe07] *Casassa Mont*, Marco; *Beato*, Filipe: On Parametric Obligation Policies: Enabling Privacy-aware Information Lifecycle Management in Enterprises, Bristol 2007, http://www.hpl.hp.com/techreports/2007/HPL-2007-7.pdf (Zugriff: 17.03.2012)

[CaPB04] *Casassa Mont*, Marco; *Pearson*, Siani; *Bramhall*, Pete: An Adaptive Privacy Management System For Data Repositories, 2004, http://www.hpl.hp.com/techreports/2004/HPL-2004-211.pdf (Zugriff: 17.03.2012)

[Casa04] *Casassa Mont*, Marco: Identity Management: On the "Identity = Data + Policies" Model, 2004, http://www.hpl.hp.com/techreports/2004/HPL-2004-14.pdf (Zugriff: 17.03.2012)

[Casa05] *Casassa Mont*, Marco: A System to Handle Privacy Obligations in Enterprises, 2005, http://www.hpl.hp.com/techreports/2005/HPL-2005-180.pdf (Zugriff: 17.03.2012)

[Cert11] *Certes Networks* (Hrsg.): TrustNet Group Encryption, 2011, http://www.certesnetworks.com/pdf/wp-group-encryption.pdf (Zugriff: 03.03.2012)

[ChHe91] *Chaum*, David; *van Heyst*, Eugène: Group Signatures; in: Advances in Cryptology - EUROCRYPT '91, Lecture Notes in Computer Science Band 547, S.257-265, Springer-Verlag, Berlin 1991

[ChJo03] *Cha*, Shi-Cho; *Joung*, Yuh-Jzer: From P3P to Data Licenses; in: *Dingledine*, Roger (Hrsg.): Privacy Enhancing Technologies, Third International Workshop, PET 2003 Dresden, Germany, Revised Papers; Lecture Notes in Computer Science Band 2760, S.205-221, Springer-Verlag, Berlin 2003

[CLFF+00] *Chan*, Nancy; *Lockwood*, Rob; *Freeman*, Stephan; *Farmah*, Pavan; *Chousiadis*, Costas; *Hamid*, Fatima; *Diedert*, Marc; *Levy*, Paul; *Hoon*, Keum; *Lewis*, Richard; *Sreeharan*, Richard; *Kamon*, Toshihiko; *Mitsianis*, John; *Das-Purkayastha*, Ari; *Chung*, Yong Wook: Denial of Service, 2000, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.8793&rep=rep1&type=pdf (Zugriff: 17.03.2012)

[Cock01] *Cocks*, Clifford: An Identity Based Encryption Scheme based on Quadratic Residues; in: Cryptography and Coding: 8th IMA International Conference Cirencester, UK, December 17-19, 2001, Proceedings, Lecture Notes in Computer Science Band 2260, S.360-363, Springer-Verlag, Berlin 2001

[CoHa09] *Cooper*, Alissa; *Hardie*, Ted: GEOPRIV: Creating Building Blocks for Managing Location Privacy on the Internet; in: IETF Journal Vol. 5 Issue 2, September 2009

[CoMN90] *McCollum*, C.J.; *Messing*, J.R.; *Notargiacomo*, L.: Beyond the pale of MAC and DAC-defining new forms of access control; in: 1990 IEEE Computer Society Symposium on Research in Security and Privacy: Conference Proceedings, IEEE Computer Society Press, Washington 1990

[Cran04] *Crane*, Stephen: Privacy Preserving Trust Agents, 2004, www.hpl.hp.com/techreports/2004/HPL-2004-197.pdf (Zugriff: 12.03.2012)

[CrCP05] *Crane*, Stephen; *Casassa Mont*, Marco; *Pearson*, Siani: On Helping Individuals to Manage Privacy and Trust, 2005, http://www.hpl.hp.com/techreports/2005/HPL-2005-53.pdf (Zugriff: 17.03.2012)

[CTBC05] *Casassa Mont*, Marco; *Thyne*, Robert; *Bramhall*, Pete; *Chan*, Kwok-Nga: Privacy Policy Enforcement in Enterprises: Addressing Regulatory Compliance and Governance Needs; in: *Sachar*, Paulus; *Pohlmann*, Norbert; *Reimer*, Helmut (Hrsg.): ISSE 2005 Securing Electronic Business Processes, Highlights of the Information Security Solutions Europe 2005 Conference, S.137-148, Vieweg, Wiesbaden 2005

[Däub02] *Däubler*, Wolfgang: Gläserne Belegschaften?, Datenschutz in Betrieb und Dienststelle, 4., überarbeitete und erweiterte Auflage; Handbücher für die Unternehmenspraxis, Bund-Verlag, Frankfurt am Main 2002

[Davi11] *Davies*, Joshua: Implementing SSL / TLS Using Cryptography and PKI, J. Wiley, New York 2011

[DDLS01] *Damianou*, Nicodemos; *Dulay*, Naranker; *Lupu*, Emil; *Sloman*, Morris: The Ponder Policy Specification Language; in: Workshop on Policies for Distributed Systems and Networks, Bristol, UK, 29-31 Jan. 2001, Lecture Notes in Computer Science Band 1995, S.18-39, Springer-Verlag, Berlin 2001

[DeLa84] *DeLaurentis*, John M.: A further weakness in the common modulus protocol for the RSA cryptoalgorithm; in: Cryptologia, Volume VIII Number 3, S.253-259, Taylor & Francis, London 1984

[Dems11] *Demsky*, Brian: Cross-application data provenance and policy enforcement; in: ACM Transactions on Information and System Security (TISSEC) Volume 14 Issue 1, ACM Press, New York 2011

[Deni09] *Denis*, Carl: Denial of Service; in: Proceedings zum Seminar Future Internet (FI) : SS2009; München 2009, http://typo3.net.in.tum.de/fileadmin/TUM/NET/NET-2009-04-1.pdf#page=40 (Zugriff: 21.02.2012)

[DiCH04] *Dillon*, Tharam S.; *Chang*, Elizabeth; *Hussain*, Farookh: A Framwork for a Trusted Environment for Virtual Collaboration; in: *Li*, Qing; *Wang*, Guoren; *Feng*, Ling (Hrsg.): Advances in Web-Age Information Management, 5th International Conference, WAIM 2004, Dalian, China, Proceedings; Lecture Notes in Computer Science Band 3129, S.1-12, Springer-Verlag, Berlin 2004

[DiGa00] *Dittrich*, Klaus R.; *Gatziu*, Stella: Aktive Datenbanksysteme, Konzepte und Mechanismen, dpunkt-Verlag, Heidelberg 2000

[DiGG96] *Dittrich*, Klaus; *Gatziu*, Stella; *Geppert*, Andreas: The Active Database Management System Manifesto: A Rulebase of ADBMS Features; SIGMOD Record, Vol. 25, No. 3, September 1996

[DiHe76] *Diffie*, Whitfield; *Hellman*, Martin E.: New Directions in Cryptography; in: IEEE Transactions on Information Theory, Vol. IT-22, No. 6 1976

[Dong09] *Dong*, Renren: Secure Multiparty Computation, Masterarbeit, Bowling Green State University, Ohio, 2009, http://etd.ohiolink.edu/send-pdf.cgi/Dong%20Renren.pdf?bgsu1241807339 (Zugriff: 20.05.2012)

[DuAt01] *Du*, Wenliang; *Atallah*, Mikhail J.: Secure Multi-Party Computation Problems and Their Applications: A Review and Open Problems; in: *Hempelmann*, Christian F.; *Raskin*, Victor (Hrsg.): Proceedings - New Security Paradigms Workshop 2001, September 10th-13th, Cloudcroft, NM, USA, S.13-22, ACM Press, New York 2001

[Ecke04] *Eckert*, Claudia: IT-Sicherheit, Konzepte - Verfahren - Protokolle, 3. Auflage, Oldenbourg, München 2004

[EEOS05] *Eckert*, Claudia; *Enzmann*, Matthias; *Okunick*, Susanne; *Schneider*, Markus: Kundenbindung durch ein anonymes Rabattsystem; in: *Horster*, Patrick (Hrsg.): D•A•CH Security 2005, Bestandsaufnahme, Konzepte, Anwendungen, Perspektiven; IT Security & IT Management, S.323-336, syssec, Klagenfurt 2005

[EPJu00] *Electronic Privacy Information Center*; *Junkbusters* (Hrsg.): Pretty Poor Privacy: An Assessment of P3P and Internet Privacy 2000, http://epic.org/reports/prettypoorprivacy.html (Zugriff: 17.03.2012)

[EsJu08] *Esch*, Martin; *Junold*, Anja: Berechtigungen in SAP ERP HCM: Konzeption, Implementierung, Betrieb, Galileo, Bonn 2008

[FeBe00] *Federrath*, Hannes; *Berthold*, Oliver: Identitätsmanagement; in: *Bäumler*, Helmut (Hrsg.): E-Privacy, Datenschutz im Internet; DuD-Fachbeiträge, S.189-204, Vieweg, Wiesbaden 2000

[Feda07] *Al-Fedaghi*, Sabah: Dismantling the Twelve Privacy Purposes; Trust management: proceedings of IFIPTM 2007, Band 238, Joint iTrust and PST Conferences on Privacy, Trust Management and Security, New Brunswick, Canada, S.207-222, Springer-Verlag, Berlin 2007

[FePf03] *Federrath*, Hannes; *Pfitzmann*, Andreas: Technische Grundlagen; in: *Roßnagel*, Alexander; *Abel*, Ralf Bernd (Hrsg.): Handbuch Datenschutzrecht, die neuen Grundlagen für Wirtschaft und Verwaltung, Beck Verlag, München 2003

[FePf11] *Federrath*, Hannes; *Pfitzmann*, Andreas: Datensicherheit; in: *Schulte*, Martin; *Schröder*, Rainer (Hrsg.): Handbuch des Technikrechts, 2. Auflage, Springer-Verlag, Berlin 2011

[FFHM+11] *Federrath*, Hannes; *Fuchs*, Karl-Peter; *Herrmann*, Dominik; *Maier*, Daniel; *Scheuer*, Florian; *Wagner*, Kai: Grenzen des "digitalen Radiergummis" in: DuD 35 (Juni 2011) S.403-407, Vieweg, Wiesbaden 2011

[FoKr05] *Forgó*, Nikolaus; *Krügel*, Tina: Die Subjektivierung der Zweckbindung, Datenschutz - Bremsblock oder Motor des E-Government?; in: DuD 29 (Dez. 2005) S.732-735, Vieweg, Wiesbaden 2005

[Fox10] *Fox*, Dirk: Computerreservierungssysteme (CRS) und Passenger Name Records (PNR); in: DuD 34 (April 2010), Vieweg, Wiesbaden 2010, sowie http://www.secorvo.de/publikationen/passenger-name-records-fox-2010.pdf (Zugriff: 17.03.2012)

[Fränkl05] *Fränkl*, Gerald: Digital Rights Management in der Praxis, Hintergründe, Instrumente, Perspektiven, (und) Mythen, Verlag Dr. Müller, Berlin 2005

[FrKA04] *Fränkl*, Gerald; *Karpf*, Philipp: Digital Rights Management Systeme, Einführung, Technologien, Recht, Ökonomie und Marktanalyse, PG-Verlag, München 2004

[Gabl12] *Gabler Wirtschaftslexikon* (Hrsg.): Stichwort: Kontrolle, Gabler Verlag, Frankfurt 2012

[Garf94] *Garfinkel*, Simson: PGP, Pretty Good Privacy, O'Reilly, Cambridge 1994

[GaSp02] *Garfinkel*, Simson; *Spafford*, Gene: Web Security, Privacy & Commerce, Security for Users, Administrators & ISPs, 2., erweiterte und aktualisierte Auflage, O'Reilly, Cambridge 2002

[Gerh09] *Gerhartl*, Andreas: Persönlichkeitsschutz im Arbeitsverhältnis, Kontrolle, Gleichbehandlung, Datenschutz; Fachbuch Recht, Linde Verlag, Wien 2009

[GKLL09] *Geambasu*, Roxana; *Kohno*, Tadayoshi; *Levy*, Amit A.; *Levy*, Henry M.: Vanish: Increasing Data Privacy with Self-Destructing Data; Proceedings of the 18th USENIX Security Symposium, Montreal, Canada, August 2009

[GNOS+04] *Gavriloaie*, Rita; *Nejdl*, Wolfgang; *Olmedilla*, Daniel; *Seamons*, Kent E.; *Winslett*, Marianne: No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web; in: Proceedings of 1st European Semantic Web Symposium (ESWS 2004), Lecture Notes in Computer Science Band 3053, S.342-356, Springer-Verlag, Berlin 2004

[GoJa01] *Gola*, Peter; *Jaspers*, Andreas: Das neue BDSG im Überblick, Erläuterungen und Schaubilder für die Datenschutzpraxis, 3. Auflage, Datakontext, Frechen 2001

[GoKl03] *Gola*, Peter; *Klug*, Christoph: Grundzüge des Datenschutzrechts, Beck Verlag, München 2003

[Gold02] *Goldberg*, Ian: Privacy-Enhancing Technologies for the Internet, II: Five Years Later; in: *Dingledine*, Roger; *Syverson*, Paul (Hrsg.): Privacy Enhancing Technologies, Second International Workshop, PET 2002 San Francisco, CA, USA, Revised Papers; Lecture Notes in Computer Science Band 2482, S.1-12, Springer-Verlag, Berlin 2002

[GoSK05] *Gola*, Peter; *Schomerus*, Rudolf; *Klug*, Christoph: BDSG – Bundesdatenschutzgesetz. Kommentar, 8., überarbeitete und ergänzte Auflage, Beck Verlag, München 2005

[Greß01] *Greß*, Sebastian: Datenschutzprojekt P3P, Darstellung und Kritik in DuD 25 (Mär. 2001) S.144-149, Vieweg, Wiesbaden 2001

[Grim03] *Grimm*, Rüdiger: Datenverarbeitung im Internet; in: *Roßnagel*, Alexander; *Banzhaf*, Jürgen; *Grimm*, Rüdiger (Hrsg.): Datenschutz im Electronic Commerce, Technik - Recht - Praxis; Schriftenreihe Kommunikation & Recht Band 18, S.21-118, Recht und Wirtschaft, Heidelberg 2003

[Grim05] *Grimm*, Rüdiger: DRM-Techniken und ihre Grenzen; in: *Picot*, Arnold; *Thielmann*, Heinz (Hrsg.): Distribution und Schutz digitaler Medien durch Digital Rights Management, S.85-96, Springer-Verlag, Berlin 2005

[GrPu06] *Grimm*, Rüdiger; *Puchta*, Stefan: Datenspuren bei der Nutzung von Digital Rights Management-Systemen (DRM) in DuD 30 (Feb. 2006) S.74-79, Vieweg, Wiesbaden 2006

[GuBh05] *Gupta*, Rajeev; *Bhide*, Manish: A Generic XACML Based Declarative Authorization Scheme for Java, Architecture and Implementation; in: *De Capitani di Vimercati*, Sabrina; *Syverson*, Paul; *Gollmann*, Dieter (Hrsg.): Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security Milan, Italy, Proceedings; Lecture Notes in Computer Science Band 3679, S.44-63, Springer-Verlag, Berlin 2005

[HaAb04] *Hayati*, Katia; *Abadi Martín*: Language-Based Enforcement of Privacy Policies; in: *Martin*, David; *Serjantov*, Andrei (Hrsg.): Privacy Enhancing Technologies, 4th International Workshop, PET 2004 Toronto, Canada, Revised Selected Papers; Lecture Notes in Computer Science Band 3424, S.302-313, Springer-Verlag, Berlin 2004

[Hans04] *Hansen*, Markus: A Double-Edged Blade, On Trusted Computing's Impact on Privacy in DuD 28 (Sep. 2004) S.525-528, Vieweg, Wiesbaden 2004

[Hans06] *Hansen*, Markus: DRM-Desaster: Das Sony BMG-Rootkit, Dubiose DRM-Software unterwandert System-Sicherheit in DuD 30 (Feb. 2006) S.95-97, Vieweg, Wiesbaden 2006

[Hans08] *Hansen*, Marit: Privacy policy languages and protocols; in: FIDIS D3.8: Study on protocols with respect to identity and identification - an insight on network protocols and privacy-aware communication (Kapitel 3.3), 2008, http://www.fidis.net/resources/deliverables/hightechid/int-d37003/ (Zugriff: 17.03.2012)

[Hatt09] *Hattendorf*, Anton: Moderne Botnetze; in: Proceedings zum Seminar Future Internet (FI) : SS2009; http://typo3.net.in.tum.de/fileadmin/TUM/NET/NET-2009-04-1.pdf#page=40 (Zugriff: 17.03.2012)

[Hauf11] *Haufe-Lexware* (Hrsg.): Datenschutz von A-Z, 2. Auflage, Haufe, Freiburg 2011

[HBSO03] *Holt*, Jason E.; *Bradshaw*, Robert W.; *Seamons*, Kent E.; *Orman*, Hilarie: Hidden Credentials; in: *Samarati*, Pierangela; *Syverson*, Paul (Hrsg.): Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society, Washington, DC, USA, S.1-8, ACM Press, New York 2003

[HiBP05] *Hilty*, Manuel; *Basin*, David; *Pretschner*, Alexander: On Obligations; in: *De Capitani di Vimercati*, Sabrina; *Syverson*, Paul; *Gollmann*, Dieter (Hrsg.): Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security Milan, Italy, Proceedings; Lecture Notes in Computer Science Band 3679, S.98-117, Springer-Verlag, Berlin 2005

[Hjel01] *Hjelm*, Johan: Creating the Semantic Web with RDF; Professional Developer's Guide Series, Wiley-VCH Verlag, New York 2001

[Horn00] *Hornberger*, Werner: Sicherheit und Datenschutz mit SAP-Systemen; SAP Press, Galileo, Bonn 2000

[IBM96] *IBM*: Building the Infrastructure for the Internet, Redbook 1996, http://www.redbooks.ibm.com/redbooks/pdfs/sg244824.pdf (Zugriff: 17.03.2012)

[IlSm02] *Iliev*, Alex; *Smith*, Sean: Prototyping an Armored Data Vault, Rights Management on Big Brother's Computer; in: *Dingledine*, Roger; *Syverson*, Paul (Hrsg.): Privacy Enhancing Technologies, Second International Workshop, PET 2002 San Francisco, CA, USA, Revised Papers; Lecture Notes in Computer Science Band 2482, S.144-159, Springer-Verlag, Berlin 2002

[JaSc04] *Janson*, Andreas; *Schwarz*, Felix: Mehrseitige Sicherheit, Sicherheitsaspekte in der Softwaretechnik, 2004, http://www.felix-schwarz.name/files/uni/Sicherheitsaspekte%20in%20der%20Softwaretechnik/Ausarbeitung%20Mehrseitige%20Sicherheit.pdf (Zugriff: 12.05.2012)

[KaGr11] *Karla*, Jürgen; *Gronenschild*, Björn: Ansatzpunkte zum Schutz personenbezogener Daten bei Nutzung von Social Media-Diensten am Arbeitsplatz; in: Lecture Notes in Informatics, GI-Edition, Proceedings Informatik 2011, Köllen Druck & Verlag, Bonn 2011

[KaPF01] *Kang*, Myong H.; *Park*, Joon S.; *Froscher*, Judith N.: Access Control Mechanisms for Inter-Organizational Workflow; in: Symposium on Access Control Models and Technologies: Proceedings of Sixth ACM Symposium on Access Control Models and Technologies, SACMAT 2001; Litton-TASC, Westfields II Conference Facility, Chantilly, Virginia, USA, S.66-74, ACM Press, New York 2001

[Karp06] *Karp*, Alan H.: Authorization-Based Access Control for the Services Oriented Architecture, 2006, http://www.hpl.hp.com/techreports/2006/HPL-2006-3.pdf (Zugriff: 17.03.2012)

[KaSW02] *Karjoth*, Günter; *Schunter*, Matthias; *Waidner*, Michael: Platform for Enterprise Privacy Practices: Privacy-Enabled Management of Customer Data; in: *Dingledine*, Roger; *Syverson*, Paul (Hrsg.): Privacy Enhancing Technologies, Second International Workshop, PET 2002 San Francisco, CA, USA, Revised Papers; Lecture Notes in Computer Science Band 2482, S.69-84, Springer-Verlag, Berlin 2002

[KaTD11] *Karamanian*, Andre; *Tenneti*, Srinivas; *Dessart*, Francois: PKI Uncovered, Certificate-Based Security Solutions for Next-Generation Networks, Macmillan Technical Publishing, Indianapolis 2011

[KCLC07] *Kumaraguru*, Ponnurangam; *Cranor*, Lorrie Faith; *Lobo*, Jorge; *Calo*, Seraphin B.: A Survey of Privacy Policy Languages; Workshop on Usable IT Security Management (USM '07), 2007, http://cups.cs.cmu.edu/soups/2007/workshop/Privacy_Policy_Languages.pdf (Zugriff: 17.03.2012)

[KhHe07] *Khadraoui*, Djamel; *Herrmann*, Francine (Hrsg.): Advances in Enterprise Information Technology Security, Information Science Reference, Hershey 2007

[KiTY04] *Kiayias*, Aggelos; *Tsiounis*, Yiannis; *Yung*, Moti: Traceable Signatures; in: *Cachin*, Christian; *Camenisch*, Jan (Hrsg.): Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Application of Cryptographic Techniques, Interlaken, Switzerland, Proceedings; Lecture Notes in Computer Science Band 3027, S.571-589, Springer-Verlag, Berlin 2004

[KiYu05] *Kiayias*, Aggelos; *Yung*, Moti: Group Signatures with Efficient Concurrent Join; in: *Cramer*, Ronald (Hrsg.): Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Application of Cryptographic Techniques, Aarhus, Denmark, Proceedings; Lecture Notes in Computer Science Band 3494, S.198-214, Springer-Verlag, Berlin 2005

[Kolt10] *Kolter*, Jan Paul: User-Centric Privacy, A Usable and Provider-Independent Privacy Infrastructure; Electronic Commerce Band 41, Josef Eul Verlag, Lohmar 2010

[KoMP01] *Koch*, M.; *Mancini*, L.V.; *Parisi-Presicce*, F.: On the Specification and Evolution of Access Control Policies; in: Symposium on Access Control Models and Technologies: Proceedings of Sixth ACM Symposium on Access Control Models and Technologies, SACMAT 2001; Litton-TASC, Westfields II Conference Facility, Chantilly, Virginia, USA, S.121-130, ACM Press, New York 2001

[KoNe04a] *Koenig*, Christian; *Neumann*, Andreas: Wettbewerbsrechtliche Aspekte vertrauenswürdiger Systemumgebungen; in: *Koenig*, Christian; *Neumann*, Andreas; *Katzschmann*, Tobias (Hrsg.): Trusted Computing; Schriftenreihe Kommunikation & Recht Band 22, S.100-140, Recht und Wirtschaft, Heidelberg 2004

[KPPK11] *Kumari*, Prachi; *Pretschner*, Alexander; *Peschla*, Jonas; *Kuhn*, Jens-Michael: Distributed data usage control for web applications: a social network implementation; in: CODASPY '11 Proceedings of the first ACM conference on Data and application security and privacy, San Antonio, TX, USA, S.85-96, ACM Press, New York 2011

[Kuhl04a] *Kuhlmann*, Dirk: Open Trusted Computing als technopolitische Herausforderung; in: *Koenig*, Christian; *Neumann*, Andreas; *Katzschmann*, Tobias (Hrsg.): Trusted Computing; Schriftenreihe Kommunikation & Recht Band 22, S.163-179, Recht und Wirtschaft, Heidelberg 2004

[KuRB08] *Kurkovsky*, Stan; *Rivera*, Oscar; *Bhalodi*, Jay: Classification of Privacy Management Techniques in Pervasive Computing; International Journal of u- and e-Service, Science and Technology Vol. 1 No. 1, Science & Engineering Research Support Center, DaeGu 2008, http://www.sersc.org/journals/IJUNESST/vol1_no1/papers/07.pdf (Zugriff: 17.03.2012)

[KüWe05] *Kühnhauser*, Winfried E.; *Welsche*, Gabriel: Sicherheitsmodelle für computergestützte Teamarbeit; in: *Horster*, Patrick (Hrsg.): D•A•CH Security 2005, Bestandsaufnahme, Konzepte, Anwendungen, Perspektiven; IT Security & IT Management, S.237-249, syssec, Klagenfurt 2005

[LAER+04] *Lefevre*, K.; *Agrawal*, Rakesh; *Ercegovac*, V.; *Ramakrishnan*, Srikant; *Xu*, Yirong; *DeWitt*, D.: Limiting Disclosure in Hippocratic Databases; in: Proc. of the 30th Int'l Conf. on Very Large Databases (VLDB 2004), Toronto, Canada 2004

[LeOS11] *Lehnert*, Volker; *Otto*, Anna; *Stelzner*, Katharina: Datenschutz in SAP-Systemen: Konzeption und Implementierung, Galileo, Bonn 2011

[LeSt09] *Lehnert*, Volker; *Stelzner*, Katharina: SAP-Berechtigungswesen: Konzeption und Realisierung, Galileo, Bonn 2009

[MaCW06] *Madsen*, Paul; *Cassasa Mont*, Marco; *Wilton*, Robin: A Privacy Policy Framework, A position paper for the W3C Workshop of Privacy Policy Negotiation, 2006, http://www.w3.org/2006/07/privacy-ws/papers/28-madsen-framework/ (Zugriff: 17.03.2012)

[MaMZ05] *Massacci*, Fabio; *Mylopoulos*, John; *Zannone*, Nicola: Minimimal Disclosure in Hierarchical Hippocratic Databases with Delegation; in: *De Capitani di Vimercati*, Sabrina; *Syverson*, Paul; *Gollmann*, Dieter (Hrsg.): Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security Milan, Italy, Proceedings; Lecture Notes in Computer Science Band 3679, S.438-454, Springer-Verlag, Berlin 2005

[Mark08] *Marko*, Kurt: A Data-Centric Security Model, Data Protection Is The Ideal Supplement To Traditional Infrastructure Security; in: PROCESSOR Vol.30 Issue 26, S.25-26, Sandhills Publishing Company, Lincoln 2008

[Mass92] *Massey*, J.L.: Contemporary Cryptology: An Introduction; in: Contemporary Cryptology: The Science of Information Integrity, S.1-39, IEEE Computer Society Press, Washington 1992

[May04] *May*, Michael J.: Summary, Analysis, and Examples of EPAL 1.2, Pennsylvania, 2004, http://wenku.baidu.com/view/5996dc6ba45177232f60a2c5.html?from=related (Zugriff: 17.03.2012)

[Mein06] *Meints*, Martin: Protokollierung bei Identitätsmanagementsystemen, Anforderungen und Lösungsansätze; in: DuD 30 (Mai 2006) S.304-307, Vieweg, Wiesbaden 2006

[Möll04] *Möller*, Jan: FAQ zum Thema Enterprise Privacy Authorization Language (EPAL), 2004, https://www.datenschutzzentrum.de/faq/epal.htm (Zugriff: 17.03.2012)

[Möll06] *Möller*, Jan: Automatisiertes Management von Datenschutzrechten, Zum Einsatz von DRM jenseits der Urheberrechte; in: DuD 30 (Feb. 2006) S.98-101, Vieweg, Wiesbaden 2006

[Moor92] *Moore*, J.H.: Protocol Failures in Cryptosystems; in: Contemporary Cryptology: The Science of Information Integrity, IEEE Computer Society Press, Washington 1992

[Münc05] *Münch*, Peter: Technisch-organisatorischer Datenschutz, Leitfaden für Praktiker, 2. überarbeitete und erweiterte Auflage; Edition IT-Sicherheit, Datakontext, Köln 2005

[MuWa10] *Murphy*, Mike; *Waterfill*, Mark: The New Hipaa Guide for 2010, Arra ACT for Hipaa Security and Compliance Law & Hitech ACT, AuthorHouse, Bloomington 2010

[NDJB02] *Nash*, Andrew; *Duane*, William; *Joseph*, Celia; *Brink*, Derek: PKI, E-Security implementieren, mitp-Verlag, Bonn 2002

[Nütz05] *Nützel*, Jürgen: Die informatorischen Aspekte virtueller Güter und Waren, Ilmenau 2005, http://juergen-nuetzel.de/virtuelle_gueter_und_waren_finale_buch_version.pdf (Zugriff: 17.03.2012)

[OECD02] *OECD*: OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, Paris 2002, http://www.oecd.org/document/18/0,3343,en_2649_34255_1815186_ 1_1_1_1,00.html (Zugriff: 17.03.2012)

[OnLZ09] *Onieva*, José; *Lopez*, Javier; *Zhou*, Jianying: Secure Multi-Party Non-Repudiation Protocols and Applications; in: Advances in Information Security Band 43, Springer-Verlag, Berlin 2009

[PaSa02] *Park*, Jaehong; *Sandhu*, Ravi: Towards Usage Control Models: Beyond Traditional Access Control; in: Symposium on Access Control Models and Technologies: Proceedings of Seventh ACM Symposium on Access Control Models and Technologies, SACMAT 2002; Naval Postgraduate School, Monterey, California, USA, S.57-64, ACM Press, New York 2002

[PaSa04] *Park*, Jaehong; *Sandhu*, Ravi: The UCONABC Usage Control Model; in: ACM transactions on information and systems security (Feb. 2004) S.128-174, ACM Press, New York 2004

[PfKö01] *Pfitzmann*, Andreas; *Köhntopp*, Marit: Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology; in: *Federrath*, Hannes (Hrsg.): Designing privacy enhancing technologies, Design Issues in Anonymity and Unobservability; Lecture notes in computer science Band 2009, S.1-9, Springer-Verlag, Berlin, Heidelberg 2001

[PoAl05] *Pons*, Alexander P.; *Aljifri*, Hassan: Active Watermarking System: Protection of Digital Media; in: *Seitz*, Juergen (Hrsg.): Digital Watermarking for Digital Media, S.233-253, Information Science Publishing, Hershey 2005

[Pond02] *Policy Research Group*: The PONDER Policy Based Management Toolkit, 2002, http://www-dse.doc.ic.ac.uk/Research/policies/ponder/PonderSummary.pdf (Zugriff: 17.03.2012)

[PoRe08] *Pohlmann*, Norbert; *Reimer*, Helmut (Hrsg.): Trusted Computing, Ein Weg zu neuen IT-Sicherheitsarchitekturen, Vieweg, Wiesbaden 2008

[Posn84] *Posner*, Richard: An economic theory of Privacy; in: *Schoeman*, F.D. (Hrsg.): Philosophical Dimensions of Privacy, S.333-345, Cambridge University Press, New York 1984

[Powe03] *Powers*, Shelley: Practical RDF, Solving Problems with the Resource Description Framework, O'Reilly, Cambridge 2003

[PRIME08] *Borking*, John et al.: Privacy and Identity Management for Europe - PRIME White Paper V3 2008, https://www.prime-project.eu/prime_products/whitepaper/PRIME-Whitepaper-V3.pdf (Zugriff: 17.03.2012)

[QiAt04] *Qin*, Li; *Atluri*, Vijayalakshmi: Concept-level Access Control for the Semantic Web; in: Workshop on XML Security: Proceedings of the ACM Workshop on XML Security 2003, Fairfax, VA, USA, S.94-103, ACM Press, New York 2004

[RaPM96] *Rannenberg*, Kai; *Pfitzmann*, Andreas; *Müller*, Günter: Sicherheit, insbesondere mehrseitige IT-Sicherheit, 1996, http://www.wiiw.de/publikationen/Sicherheitinsbesonderemehrsei.pdf (Zugriff: 12.05.2012)

[Reim04] *Reimer*, Helmut: "PRIME - Privacy and Identity Management for Europe" in: DuD 28 (Mai 2004) S.321-322, Vieweg, Wiesbaden 2004

[Rijm10] *Rijmenants*, Dirk: Is One-time Pad History?; in: Cipher Machines and Cryptology 2010, http://users.telenet.be/d.rijmenants/papers/is_one_time_pad_history.pdf (Zugriff: 17.03.2012)

[Rijm10a] *Rijmenants*, Dirk: The complete guide to secure communications with the one time pad cipher; in: Cipher Machines and Cryptology 2010, http://users.telenet.be/d.rijmenants/papers/one_time_pad.pdf (Zugriff: 17.03.2012)

[Roes00] *Roessler*, Thomas: Vermeidung von Spuren im Netz; in: *Bäumler*, Helmut (Hrsg.): E-Privacy, Datenschutz im Internet; DuD-Fachbeiträge, S.205-213, Vieweg, Wiesbaden 2000

[RoPG01] *Roßnagel*, Alexander; *Pfitzmann*, Andreas; *Garstka*, Hansjürgen: Modernisierung des Datenschutzrechts, Gutachten im Auftrag des Bundesministeriums des Innern, Müller-Druck, Berlin 2001

[RoTM02] *Rosenblatt*, Bill; *Trippe*, Bill; *Mooney*, Stephen: Digital Rights Management, Business and Technology, M&T Books, New York 2002

[RSA78] *Rivest*, Ronald L.; *Shamir*, Adi; *Adleman*, Leonard: A method for obtaining digital signatures and public-key cryptosystems; in: Communications of the ACM, v.21, No. 2, S.120-126, 1978

[Sael04] *Saeltzer*, Gerhard: Sind diese Daten personenbezogen oder nicht?; in: DuD 28 (Apr. 2004) S.218-227, Vieweg, Wiesbaden 2004

[Sant09] *dos Santos Cardoso*, André: Originator Controlled Access Control, 2009, http://web.fe.up.pt/~jmcruz/ssi/ssi.0910/trabs-als/apres.7-andre.cardoso.pdf (Zugriff: 17.03.2012)

[SaPa03] *Sandhu*, Ravi; *Park*, Jaehong: Usage Control: A Vision for Next Generation Access Control; in: *Gorodetsky*, Vladimir; *Popyack*, Leonard; *Skormin*, Victor (Hrsg.): Computer Network Security, Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2003, St. Petersburg, Russia, Proceedings; Lecture Notes in Computer Science Band 2776, S.17-31, Springer-Verlag, Berlin 2003

[Scha11] *Schaar*, Peter: Tätigkeitsbericht 2009 und 2010 des Bundesbeauftragten für den Datenschutz und die Informationsfreiheit – 23. Tätigkeitsbericht –, Bundestagsdrucksache 17/5200, Druck+Verlag, Bonn 2011

[Schm09] *Schmeh*, Klaus: Kryptografie, Verfahren, Protokolle, Infrastrukturen, 4., aktualisierte und erweiterte Auflage, dpunkt-Verlag, Heidelberg 2009

[Schn96] *Schneier*, Bruce: Angewandte Kryptographie, Protokolle, Algorithmen und Sourcecode in C, Addison-Wesley, Bonn 1996

[Schr01] *Schreck*, Jörg: Security and Privacy in User Modeling, Essen 2001, http://www.jschreck.de/sapium/ (Zugriff: 17.03.2012)

[ScTW04] *Schmidt*, Andreas U.; *Tafreschi*, Omid; *Wolf*, Ruben: Interoperability Challenges for DRM Systems, Ilmenau 2004, http://virtualgoods.tu-ilmenau.de/2004/Interoperability_Challenges_for_DRM_Systems.pdf (Zugriff: 17.03.2012)

[Selk03] *Selk*, Robert: Datenschutz und Internet, Das TDDSG und dessen Rolle für ein modernes Datenschutzrecht, dargestellt anhand der Technik der Cookies, Der Andere Verlag, Osnabrück 2003

[SeMa04] *Serrao*, Carlos; *Marques*, Joaquim: Enabling Digital Content Protection on Super-Distribution Models, Ilmenau 2004, http://virtualgoods.tu-ilmenau.de/2004/VG2004-EDCP-SD-OSDRM.pdf (Zugriff: 17.03.2012)

[Sham79] *Shamir*, Adi: How to Share a Secret; in: Communications of the ACM vol. 22, S.612-613, ACM Press, New York 1979, sowie http://groups.csail.mit.edu/cis/crypto/classes/6.857/papers/secret-shamir.pdf (Zugriff: 17.03.2012)

[Sham84] *Shamir*, Adi: Identity-based cryptosystems and signature schemes; in: Advances in Cryptology - Crypto '84, Lecture Notes in Computer Science Band 196, S.47-53, Springer-Verlag, Berlin 1984

[SHPR+06] *Sandeep*, Bhatt; *Horne*, William; *Pato*, Joe; *Rajagopalan*, S. Raj; *Rao*, Prasad: Model-based validation of enterprise access policies, 2006, http://www.hpl.hp.com/techreports/2005/HPL-2005-152R1.pdf (Zugriff: 17.03.2012)

[ShSB08] *Shah*, Mehul A.; *Swaminathan*, Ram; *Baker*, Mary: Privacy-Preserving Audit and Extraction of Digital Contents, 2008, http://www.hpl.hp.com/techreports/2008/HPL-2008-32R1.pdf (Zugriff: 17.03.2012)

[SiBW95] *Sibert*, Olin; *Bernstein*, David; *Van Wie*, David: The DigiBox: A Self-Protecting Container for Information Commerce; in: USENIX Association (Hrsg.): The First USENIX Workshop on Electronic Commerce, Conference Proceedings, S.171-183, New York 1995

[Simm83] *Simmons*, Gustavus James: A "Weak" Privacy Protocol Using the RSA Crypto Algorithm; in: Cryptologia, Volume VII Number 2, 1983, S.180-182, Taylor & Francis, London 1983

[SkBC04] *Skogsrud*, Halvard; *Benatallah*, Boualem; *Casati*, Fabio: Trust-Serv: Model-Driven Lifecycle Management of Trust Negotiation Policies for Web Services, 2004, http://www.hpl.hp.com/techreports/2004/HPL-2004-41.pdf (Zugriff: 17.03.2012)

[Smar11] *Smart*, Nigel (Hrsg.): ECRYPT II Yearly Report on Algorithms and Keysizes, Leuven 2011, http://www.ecrypt.eu.org/documents/D.SPA.17.pdf (Zugriff: 17.03.2012)

[Sofi04] *Sofiotis*, Ilias: Der Zusammenhang zwischen dem Recht auf informationelle Selbstbestimmung und dem Recht auf Informationszugang, erläutert anhand der Problematik der Informationshilfe, Unter Berücksichtigung der Vorgaben des Europäischen Gemeinschaftsrechts zum Datenschutz und zum Informationszugang, o.V., Köln 2004

[SpPo11] *Spogahn*, Nikolai; *Pohlmann*, Norbert: Was Google alles über jeden wissen könnte; in: iX 7/2011, Heise Verlag, Hannover 2011

[STMC+07] *Schulzrinne*, Henning; *Tschofenig*, Hannes; *Morris*, John B.; *Cuellar*, Jorge R.; *Polk*, James; *Rosenberg*, Jonathan: RFC4745: Common Policy: A Document Format for Expressing Privacy Preferences, 2007, http://www.ietf.org/rfc/rfc4745.txt (Zugriff: 17.03.2012)

[TaWe06] *Talbot*, John; *Welsh*, Dominic: Complexity and Cryptography, An Introduction, Cambridge University Press, New York 2006

[ThSa97] *Thomas*, R. K.; *Sandhu*, R.S.: Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management; in: Proceedings of the IFIP WG11.3 Workshop on Database Security, Lake Tahoe, California, S.166-181, Chapman & Hall, London 1997

[Tilb05] *van Tilborg*, Henk C.A. (Hrsg.): Encyclopedia of Cryptography and Security, Springer-Verlag, New York 2005

[Trce06] *Trcek*, Denis: Managing informatiom systems security and privacy, Springer-Verlag, Berlin 2006

[Ulbr11] *Ulbricht*, Max-R.: Regulierung von Nutzerverhalten innerhalb sozialer Netzwerke, - Facebook und die Privatsphäre; in: Lecture Notes in Informatics, GI-Edition, Proceedings Informatik 2011, Köllen Druck & Verlag, Bonn 2011

[ULD11] *Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein* (Hrsg.): Tracking - Nutzerverfolgung im Internet, 2011, https://www.datenschutzzentrum.de/tracking/ (Zugriff: 17.03.2012)

[Venk07] *Venkataramu*, Ramya: Analysis and Enhancement of Apple's FairpPlay Digital Rights Management, Masterarbeit, San Jose State University 2007

[Wagn11] *Wagner*, Kai: Schlüsselaustausch und Policy Enforcement bei zweckgebundener Datenübermittlung; in: Lecture Notes in Informatics, GI-Edition, Proceedings Informatik 2011, Köllen Druck & Verlag, Bonn 2011[277]

[Weic00] *Weichert*, Thilo: Zur Ökonomisierung des Rechts auf informationelle Selbstbestimmung; in: *Bäumler*, Helmut (Hrsg.): E-Privacy, Datenschutz im Internet; DuD-Fachbeiträge, S.158-184, Vieweg, Wiesbaden 2000

[277] Aus dieser Dissertation hervorgegangene Veröffentlichung.

[Zieg02] *Ziegler*, Cai: Zankapfel, P3P: die Platform for Privacy Preferences des W3C in iX 3/ 2002, S.129ff, Heise Verlag, Hannover 2002

[Zimm01] *Zimmermann*, Jürgen: Konzeption und Realisierung eines aktiven Datenbanksystems: Architektur, Schnittstellen und Werkzeuge, Logos, Berlin 2001

[ZPPS04] *Zhang*, Xinwen; *Park*, Jaehong; *Parisi-Presicce*, Francesco; *Sandhu*, Ravi: A Logical Specification for Usage Control; in: Symposium on Access Control Models and Technologies: Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies, SACMAT 2004, Yorktown Heights, New York, USA, S.1-10, ACM Press, New York 2004

[ZPPS05] *Zhang*, Xinwen; *Park*, Jaehong; *Parisi-Presicce*, Francesco; *Sandhu*, Ravi: Formal Model and Policy Specification of Usage Control; in: ACM transactions on information and systems security (Nov. 2005) S.351-387, ACM Press, New York 2005