Maßgeschneiderte nutzbarkeitserhaltende Pseudonymisierung

Anforderungen, Beschreibung, Umsetzung

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Raedah Saffija Kasem-Madani
aus
Bonn

Bonn 2022

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter: Prof. Dr. Michael Meier
Gutachter: Prof. Dr. Steffen Wendzel

Tag der Promotion: 21.10.2022
Erscheinungsjahr: 2023

Danksagung

Die Anfertigung der Dissertation bereitete mir viel Freude und lehrte mich Einiges. Bis zur Einreichung war jedoch ein recht steiniger Weg mit vielen Abzweigungen zu gehen. Auf diesem Weg haben mich viele Menschen begleitet und neue Menschen sind mir begegnet. Ohne die Impulse all jener Menschen wäre diese Arbeit nicht das geworden, was sie nun ist. An dieser Stelle möchte ich mich daher bei ihnen allen bedanken.

Ich danke Herrn Prof. Dr. Michael Meier herzlich für die Aufnahme in die Arbeitsgruppe IT-Sicherheit und die Möglichkeit, hier mit viel Wahlfreiheit zu forschen. Auch danke ich ihm für die Betreuung dieser Arbeit, die tiefgehenden Diskussionen und das Teilen von Erfahrungen und Erkenntnissen. Ein herzlicher Dank geht an Prof. Dr. Steffen Wendzel für die Übernahme der Rolle des Zweitgutachters und die erkenntnisreichen Diskussionen zu Themen der IT-Sicherheit und des technischen Datenschutzes auch während seiner Zeit in Bonn.

Ein herzliches Dankeschön für die bereichernden Diskussionen und die intensive, tolle Zusammenarbeit an Dr. Felix Boes, Dr. Timo Malderle, Daniel Meyer, Markus Krämer, Sebastian Karhoff, Dr. Hajira Jabeen, Sebastian Land und Gunnar Grasshoff. Für das Korrekturlesen von Teilen dieser Arbeit danke ich Dr. Felix Boes, Dr.-Ing. Sandra Loch-Dehbi, Daniel Meyer, Markus Krämer, Mariam Abd El Fatah und Mourad Madani.

Danke an Prof. Dr.-Ing. Kerstin Lemke-Rust für das Heranführen an Themen der IT-Sicherheit und die Bestärkung des Wunsches zu forschen.

Ich danke meiner Familie und meinen Freunden für ihre Begleitung, ihren Beistand, ihre Ermutigungen und ihre Geduld mit mir insbesondere in der Endphase der Dissertation. Ein besonderer Dank gilt meinem Ehemann Mourad Madani, unseren Kindern und meiner Schwester Mariam Abd El Fatah. Ohne euren Glauben an mich wäre diese Dissertation nicht möglich geworden. Diese Dissertation ist euch gewidmet.

Kurzfassung

Die Verarbeitung personenbezogener Daten ist omnipräsent. Um die Privatsphäre und die informationelle Selbstbestimmung der Betroffenen zu achten, ist das Ergreifen von Maßnahmen zum Schutze der Vertraulichkeit der Daten erforderlich. Hierzu gehört u.a. die Pseudonymisierung [55, 136].
Bisher benötigen Anwender für die Nutzung von Pseudonymisierungsverfahren Expertenwissen der Privatsphäre schützenden Techniken (engl. Privacy-Enhancing-Technologies, PET). PET sind Techniken, die u.a. auf kryptographischen Verfahren und weiteren technisch-organisatorischen Schutzmaßnahmen basieren. Dieses Wissen ist zusätzlich zum Expertenwissen im Bereich des Anwendungsgebiets erforderlich. Es wird angenommen, dass dies eine Hürde darstellt, die eine korrekte und sinnhafte Nutzung von Pseudonymisierungsverfahren durch Anwender erschwert.
In dieser Dissertation wird die grundlegende Fragestellung untersucht, wie eine effektive, nutzbarkeitserhaltende Pseudonymisierung von personenbezogenen Daten für breite Anwendergruppen zugänglich und damit praxistauglich gemacht werden kann. Es wird ein Rahmenwerk erarbeitet, mit dem Pseudonymisierungen ohne Expertenkenntnisse ganz bestimmte Nutzbarkeiten erhaltend, gleichzeitig jedoch das Risiko der Reidentifizierung Betroffener reduzierend erstellt und verarbeitet werden können.
Das Rahmenwerk besteht aus den folgenden vier Komponenten: einem Anforderungsmodell für die Definition von Nutzbarkeits- und Vertraulichkeitsanforderungen, einer Beschreibungssprache für die maschinenlesbare Anforderungsbeschreibung, einer Datenstruktur für Pseudonymisierungen und Übersetzungsregeln für die Ableitung von maßgeschneiderten Pseudonymisierungen. Das Rahmenwerk soll unabhängig vom Anwendungsfall die Erstellung von Pseudonymisierungen erleichtern. Die Sinnhaftigkeit des Einsatzes des Rahmenwerks wird innerhalb von zwei Anwendungsbeispielen veranschaulicht.

Inhaltsverzeichnis

1 Einleitung

Personenbezogene Daten werden in den unterschiedlichsten Kontexten verarbeitet. In der Medizinforschung werden personenbezogene Gesundheitsdaten gesammelt und analysiert. Beim Abrufen von Webseiten im World Wide Web werden u.a. Cookies mit Informationen zum Verhalten des Internetnutzers generiert und ausgewertet. Moderne Kraftfahrzeuge beinhalten Ortungsdienste, die das Ermitteln der Position des Fahrzeugs über Smartphone-Apps erlauben. Um die Privatsphäre und die informationelle Selbstbestimmung der Betroffenen bei der Datenverarbeitung zu achten, ist ein sorgfältiger Umgang und das Ergreifen von Maßnahmen zum Schutz der Vertraulichkeit der Daten erforderlich. Diese Anforderung ergibt sich nicht zuletzt aus geltenden Gesetzen und Verordnungen, wie etwa dem deutschen Bundesdatenschutzgesetz (BDSG) [34] und der europäischen Datenschutzgrundverordnung (DSGVO) [55]. Gleichzeitig sollen aber Wertschöpfungen und Erkenntnisgewinne aus den Daten ermöglicht werden. Dies erfordert die Verfügbarkeit von personenbezogenen Daten.

Die Vertraulichkeit und die Verfügbarkeit personenbezogener Daten stehen als Schutzziele der IT-Sicherheit in inhärentem Konflikt zueinander [129]. Um beiden Schutzzielen gerecht zu werden, müssen Maßnahmen ergriffen werden, die diesen Konflikt zumindest reduzieren. Diese Maßnahmen umfassen u.a. die Anonymisierung und die Pseudonymisierung [136]. Letztere wird im Folgenden Datenpseudonymisierung genannt. Anonymisierungsverfahren verändern die Daten derart, dass Analysen, bei denen es auf die Eigenschaften einzelner Werte der Datensammlung ankommt, nicht mehr durchgeführt werden können. Datenpseudonymisierung kann zur Minderung des Risikos der Reidentifizierung Betroffener immer dann eingesetzt werden, wenn im Rahmen der geplanten Verarbeitung eine Anonymisierung zu einem nicht vertretbaren Verlust der Nutzbarkeit der Daten führen würde. Zum Schutz der Betroffenenrechte fordert die DSGVO eine zeitnahe Datenpseudonymisierung personenbezogener Daten. Der Zeitpunkt der Datenpseudonymisierung spielt aus technischer Sicht eine entscheidende Rolle: Je frühzeitiger in der Verarbeitungskette die Daten pseudonymisiert werden, desto eher greift die Schutzfunktion [136] durch Informationsreduktion. Je nach Art der eingesetzten Pseudonymisierungsverfahren hat die Informationsreduktion jedoch einen zumindest teilweisen Verlust der Nutzbarkeit der Daten zur Folge.

Zu den Herausforderungen einer effektiven Umsetzung von Datenpseudonymisierung gehört daher die Auswahl und Anwendung geeigneter Pseudonymisierungsverfahren. Zum einen soll die geplante Anwendung auf den pseudonymisierten Daten weiterhin möglich sein. Zum Beispiel sollten Berechnungen einer Datenanalyse auch nach der Datenpseudonymisierung geeignete Berechnungsergebnisse liefern. Dies setzt eine Verfügbarkeit von Information aus den Klartextdaten voraus. Zum anderen sollte die Pseudonymisierung effektiv sein: Eine Ermittlung der zugrundeliegenden Klartextdaten aus der Pseudonymisierung sollte nur unter bestimmten

Bedingungen möglich sein. Dies wiederum setzt die Vertraulichkeit der Klartextdaten voraus. Die Verfügbarkeit und Vertraulichkeit der personenbezogenen Klartextdaten muss daher geeignet ausbalanciert werden. Weiterhin sollte die Pseudonymisierung für Anwender ohne Expertenkenntnisse praktikabel erstellbar und verarbeitbar sein.
Aktuell erfordert der Einsatz von Pseudonymisierungsverfahren durch Anwender jedoch Expertenwissen der Privatsphäre schützenden Techniken (Privacy-Enhancing-Technologies, PET) [28]. Um zum Beispiel bestimmte Nutzbarkeiten erhalten zu können, sind genaue Kenntnisse über die Eigenschaften der Chiffrate bestimmter Verschlüsselungsverfahren erforderlich. Auch ist aus Gründen der Sicherheit genaue Kenntnis des Stands der Wissenschaft und geeignete Parametrisierungen der Verfahren erforderlich. Hinzukommt, dass bisher keine standardisierten Verfahren zur nutzbarkeitserhaltenden Pseudonymisierung existieren. Bisher müssen Verfahren, die einzelne Nutzbarkeiten erhalten vom Pseudonymisierenden aus einer Vielzahl von kryptographischen Verfahren ausgewählt und umgesetzt werden. Hierbei muss für jedes der infrage kommenden Verfahren geprüft werden, ob es für die Umsetzung der geplanten Nutzbarkeit geeignet ist, dem Stand der Wissenschaft entspricht, einen ausreichenden Vertraulichkeitsschutz bietet und gleichzeitig ausreichend praktikabel ist.
Sollen im Rahmen eines beispielhaften medizinischen Experiments in Vitaldaten signifikante Abweichungen von Durchschnittswerten eines Attributs ermittelt werden, so können diese Berechnungen auf geeignet pseudonymisierten Daten ohne Kenntnis der Klartextdaten durchgeführt werden. In einer zweiten Stufe des Experiments sollen die von den abweichenden Werten Betroffenen über die errechneten Abweichungen informiert werden. Hierfür müssen die Betroffenen identifiziert werden können. Die Pseudonymisierung muss also in diesen Fällen aufgehoben werden können. Ein weiteres Beispiel ist die Auswertung des Fahrverhaltens durch Versicherungsunternehmen im Rahmen des Anbietens von Telematik-Tarifen[1]. Zum einen möchte der Versicherer die Versicherten durch seine Tarifgestaltung zu vorausschauendem, umweltfreundlichem und risikoarmen Fahren ermutigen. Um das Fahrverhalten zu überprüfen, möchte er Daten, die im Fahrzeug erhoben werden, geeignet analysieren und auswerten. Zum anderen soll der Versicherer aus den erhobenen Daten keine Kenntnis über weitere Lebensgewohnheiten der Fahrerin erhalten. Diese und ähnliche Anforderungen stellen Forschende, die personenbezogene Daten auf bestimmte Eigenschaften analysieren wollen, vor eine nur schwer überwindbare Herausforderung.

In dieser Dissertation soll die **grundlegende Fragestellung** untersucht werden, wie eine effektive, nutzbarkeitserhaltende Pseudonymisierung von personenbezogenen Daten für breite Anwendergruppen zugänglich und damit praxistauglich gemacht werden kann. In dem vorliegenden Kapitel wird daher die Erarbeitung eines **Rahmenwerks** für die nutzbarkeitserhaltende Datenpseudonymisierung motiviert und anhand von zu beantwortenden Forschungsfragen hergeleitet. Es wird im Folgenden skizziert, mit welchen **Komponenten** das erarbeitete Rahmenwerk die Forschungsfragen beantwortet. Die im Rahmen der Forschungsarbeit veröffentlichten **Publikationen** werden gelistet und den Forschungsfragen und den Komponenten des Rahmenwerks zugeordnet. **Grundlegende**

[1] Siehe z.B. den Telematiktarif der ADAC [2].

Begriffe aus den relevanten Forschungsgebieten des Datenschutzes und der Privacy sowie der Informationssicherheit und Privacy-Enhancing-Technologies werden erläutert.

1.1 Ziele und Forschungsfragen

Die vorliegende Arbeit untersucht die automatisierte Erstellung und Verarbeitung nutzbarkeitserhaltender Pseudonymisierungen.

> Ziel ist die Erarbeitung eines Rahmenwerks, mit dem ohne Expertenkenntnisse Pseudonymisierungen erstellt und verarbeitet werden können, die bestimmte Nutzbarkeiten erhalten und das Risiko der Reidentifizierung Betroffener reduzieren.

Insgesamt soll ein möglichst allgemeines Rahmenwerk erarbeitet werden, welches unabhängig vom Anwendungsfall die Erstellung von Pseudonymisierungen erlaubt. Um dieses Ziel zu erreichen, werden die folgenden Forschungsfragen untersucht:

1. Wie können **Anforderungen**, die aus der intendierten späteren Nutzung der Daten (Nutzbarkeitsanforderungen) stammen, vor der Pseudonymisierung dieser Daten formuliert werden?
2. Wie können für Nutzbarkeitsanforderungen **maßgeschneiderte Pseudonymisierungen** erstellt werden? Wie sehen geeignete Pseudonymisierungsverfahren aus?
3. Wie kann eine **Formulierung von Nutzbarkeitsanforderungen** für eine automatisierte Erstellung von maßgeschneiderten Pseudonymisierungen verwendet werden?
4. Wie kann eine **Datenstruktur** aussehen, die eine Haltung und Verarbeitung von automatisiert erstellten, maßgeschneiderten Pseudonymisierungen unter Berücksichtigung von Vertraulichkeitsanforderungen erleichtert bzw. erst ermöglicht?

Um den **praktischen Nachweis** der Sinnhaftigkeit des an Nutzbarkeitsanforderungen orientierten Ansatzes der Forschungsfragen nachzuweisen, sollen exemplarisch Anwendungen an ausgewählten Beispielen erarbeitet werden.

1.2 Rahmenwerk

Zur Beantwortung der Forschungsfragen dieser Arbeit wird ein Rahmenwerk für die nutzbarkeitserhaltende Pseudonymisierung semistrukturierter Daten entwickelt. Dieses Rahmenwerk besteht aus den folgenden vier Komponenten:

1. **Anforderungsmodell:** Eine Definition von zwei grundlegenden Anforderungsklassen: Nutzbarkeitsanforderungen und Vertraulichkeitsanforderungen. Jede der beiden Klassen enthält Anforderungen an Pseudonymisierungen, die bei der Erstellung und Verarbeitung dieser beachtet werden müssen.

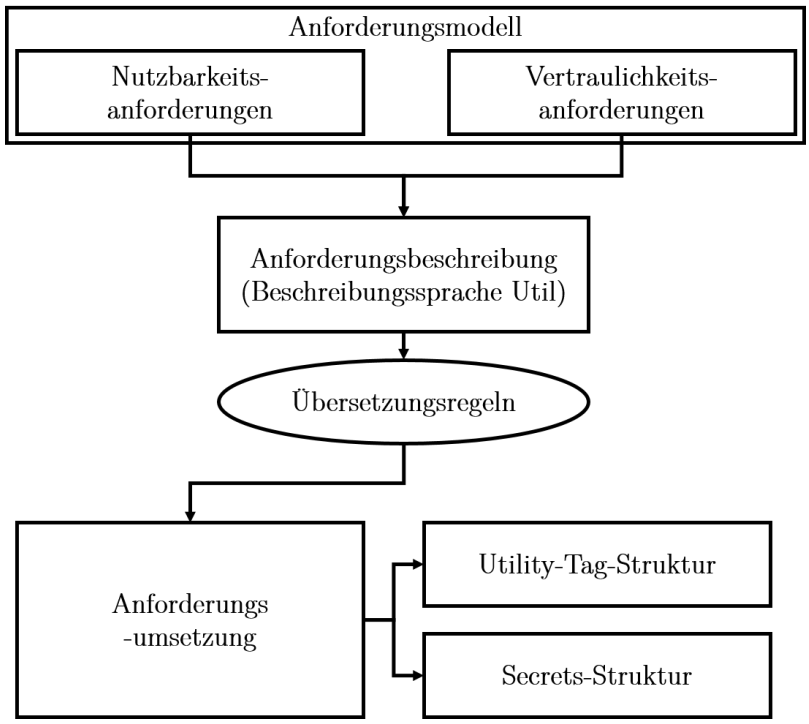

Abbildung 1: *Rahmenwerk für die automatisierte, maßgeschneiderte und nutzbarkeitserhaltende Datenpseudonymisierung.*

2. **Anforderungsbeschreibung:** Definition von `Util` als eine menschen- und maschinenlesbare Beschreibungssprache zur Formulierung von Nutzbarkeitspolitiken. Die Anwendung dieser Sprache erlaubt die Formulierung von Nutzbarkeitspolitiken als Mengen von Nutzbarkeits- und Vertraulichkeitsanforderungen an zu erstellende Pseudonymisierungen. Für die Formulierung der Anforderungen reichen Anwenderkenntnisse aus. Expertenwissen aus den Bereichen Informationssicherheit, Datenschutz, Datenschutzfördernde Techniken oder Angewandter Kryptographie ist nicht erforderlich.
3. **Utility-Tag- und Secrets-Struktur:** Eine flexible Datenstruktur für die Erstellung, Speicherung und Verarbeitung von maßgeschneiderten Pseudonymisierungen für bestimmte Nutzbarkeits- und Vertraulichkeitsanforderungen.
4. **Übersetzungsregeln:** Übersetzungsregeln für die automatisierte Umsetzung von in `Util` formulierten Anforderungen in maßgeschneiderte und somit adäquate Pseudonymisierungen mit Utility-Tags.

Die Zusammenhänge der Komponenten des Rahmenwerks sind in Abbildung 1 skizziert. Der Nachweis der Sinnhaftigkeit des Ansatzes in der Praxis erfolgt durch die Evaluierung des Rahmenwerks innerhalb von zwei Anwendungsbeispielen. Mit den Beispielen wurde die grundlegende Praxistauglichkeit des Ansatzes demonstriert. Die beiden Beispiele werden in Kapitel 7 beschrieben.

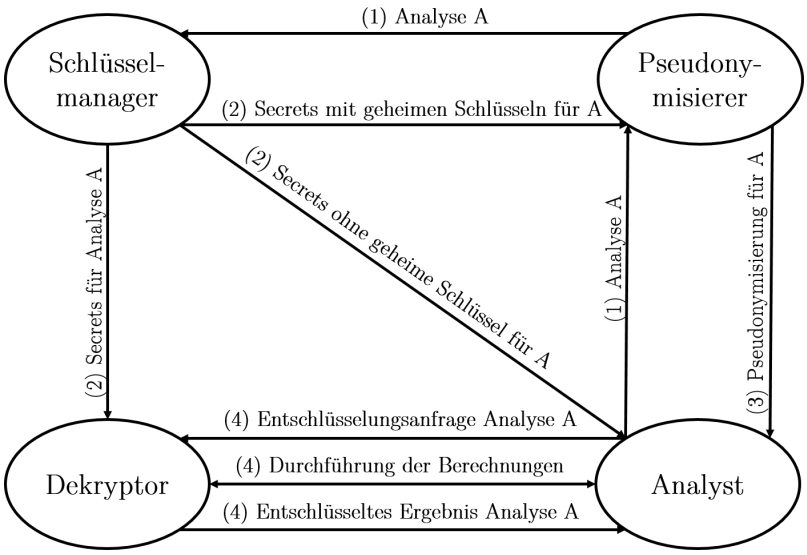

Abbildung 2: *Beispielumgebung für die Umsetzung der nutzbarkeitserhaltenden Pseudonymisierung.*

1.2.1 Akteure

Ein das Rahmenwerk umsetzendes System besteht aus unterschiedlichen Akteuren. Diese interagieren im Rahmen der intendierten Verarbeitung der Pseudonymisierungen. Diese Rollen umfassen den Pseudonymisierer, den Schlüsselmanager, den Analyst und den Dekryptor. Der Pseudonymisierer kennt die Klartextdaten. Er kann sie z.B. von kollaborierenden Dritten erhalten haben. Er erzeugt Pseudonymisierungen mithilfe von notwendiger öffentlicher oder privater Zusatzinformation (Secrets). Der Schlüsselmanager verwaltet die Secrets und leitet diese an die beteiligten Akteure weiter. Er kann mit dem Pseudonymisierer identisch sein. Der Analyst fragt Pseudonymisierungen für bestimmte Analysen beim Pseudonymisierer an. Dieser erstellt die maßgeschneiderte Pseudonymisierung und liefert sie an den Analyst aus. Der Analyst führt die Verarbeitungsschritte der geplanten Analyse durch. Hierzu nutzt er öffentliche Secrets. Ist der Zugriff auf geheime Secrets erforderlich, so interagiert der Analyst mit dem Dekryptor gemäß festgelegter Protokolle. Der Dekryptor hat Zugriff auf die öffentlichen und geheimen Secrets. In dieser Arbeit werden die genannten Akteure angenommen. In Abbildung 2 ist dieser Ansatz skizziert.

1.2.2 Angreifermodell und Sicherheitsannahmen

Für ein System, auf dem das Rahmenwerk implementiert und ausgeführt wird, werden Annahmen für eine sichere Umsetzung getroffen. Hauptziel ist hierbei die Erfüllung des Schutzziels der Vertraulichkeit. Die zu verarbeitenden personenbezogenen Klartextdaten sollen so vor unautorisierter Aufdeckung geschützt werden. Weitere Schutzziele wie die Integrität und die Authentizität spielen bei der Betrachtung der Umsetzung des Rahmenmodells eine untergeordnete Rolle. Die Annahmen werden im Folgenden beschrieben.

Verschlüsselungsverfahren Die eingesetzten Verschlüsselungsverfahren entsprechen dem Stand der Wissenschaft. Es wird angenommen, dass sie bei gleichzeitiger Berücksichtigung der in Kapitel 3 vorgestellten Anforderungen den aktuell höchstmöglichen Vertraulichkeitsschutz bieten. Wie in Kapitel 5 bei der Auswahl der Pseudonymisierungsverfahren ausgeführt wird, kann über die Nutzbarkeit hinaus unintendiert Information abfließen. Diese wird nach Möglichkeit eingeschränkt.

Vertrauenswürdigkeit Der Dekryptor, der Pseudonymisierer und der Schlüsselmanager sind vertrauenswürdig. Insbesondere kollaborieren sie nicht mit dem Analyst zur Ermittlung von Klartextdaten.

Monitoring und Zugriffskontrolle Um einen unautorisierten Abfluss von Information durch die Nutzbarkeit von Pseudonymisierungen zu erschweren, muss der Zugriff auf pseudonymisierte Daten kontrolliert und beschränkt werden. Es wird angenommen, dass zur Durchsetzung auf dem System geeignete Zugriffskontroll- und Monitoringmechanismen implementiert sind.

Protokollkonformes Verhalten der Akteure Alle Akteure verhalten sich protokollkonform. Insbesondere weichen sie nicht durch unerlaubte Aufdeckung von Klartextdaten vom Protokoll ab.

Möglichkeit des Abfliessens von Daten In der Realität implementierte Systeme sind anfällig für das unautorisierte Abfließen von Daten infolge von Angriffen[2]. Es wird in dieser Dissertation angenommen, dass bis auf einzelne, unter bestimmten Voraussetzungen aufgedeckte Klartextdaten ausschließlich pseudonymisierte Daten verarbeitet werden. Konstruktionsbedingt soll somit das Risiko einer Reidentifizierung Betroffener gemindert und eventuelle Folgen von Datenlecks abgemildert werden.

1.3 Aufbau der Dissertation

In der Dissertation wird ein Rahmenwerk zur nutzbarkeitserhaltenden Datenpseudonymisierung ohne Expertenkenntnisse beschrieben. In dem vorliegenden Kapitel werden die Grundlagen und grundlegenden Annahmen erarbeitet, die für die Überlegungen und Herleitung des Rahmenwerks erforderlich sind. Kapitel 2 beinhaltet einen Überblick über das erforschte und erarbeitete Rahmenwerk und seine Komponenten. In Kapitel 3 werden die beiden Anforderungsklassen der Nutzbarkeitsanforderungen und der Vertraulichkeitsanforderungen kategorisiert, motiviert und beschrieben. In Kapitel 4 wird mit `Util` eine Beschreibungssprache für die maschinenlesbare Formulierung von Nutzbarkeits- und Vertraulichkeitsanforderungen hergeleitet, motiviert und beschrieben. Für die Umsetzung von Nutzbarkeitsanforderungen wurden in Kapitel 5 nutzbarkeitserhaltende Pseudonymisierungsverfahren beschrieben. Diese basieren auf nach dem Stand der

[2] Für eine Übersicht aktueller Datenlecks siehe z.B. `https://www.securitymagazine.com/articles/96667-the-top-data-breaches-of-2021`

Wissenschaft sicheren, gleichzeitig aber im Vergleich mit ähnlichen Verfahren hinsichtlich Speicherplatz und Laufzeit praktikablen Verfahren. Um die Umsetzung von Nutzbarkeitsanforderungen in geeignete, nach dem Stand der Wissenschaft passend parametrisierte Pseudonymisierungsverfahren automatisieren zu können, wurden in Kapitel 6 eine Pseudonymisierungsstruktur zur Verarbeitung der Pseudonymisierung und Übersetzungsregeln zur Umsetzung von in `Util` formulierten Anforderungen in geeignete Pseudonymisierungsverfahren entwickelt. Um die Sinnhaftigkeit des Ansatzes zu illustrieren, wurden in Kapitel 7 für zwei Anwendungsbeispiele ausgeführt, wie das Rahmenwerk zur Datenpseudonymisierung genutzt werden kann. Die Ergebnisse der Arbeit, ein Fazit und ein Ausblick auf vielfältige, auf Basis der Arbeit zu bearbeitende Forschungsthemen werden in Kapitel 8 zusammengefasst.

1.4 Veröffentlichungen

Das grundlegende Konzept des Rahmenwerks, dessen Komponenten und ein Anwendungsbeispiel wurden in den folgenden Arbeiten veröffentlicht:

Kasem-Madani S., Meier M., Wehner M. Towards a Toolkit for Utility and Privacy-Preserving Transformation of Semi-structured Data Using Data Pseudonymization. In: Garcia-Alfaro J., Navarro-Arribas G., Hartenstein H., Herrera-Joancomartí J. (eds) Data Privacy Management, Cryptocurrencies and Blockchain Technology. ESORICS 2017, DPM 2017, CBT 2017. Lecture Notes in Computer Science, vol 10436. Springer, Cham. 2017. https://doi.org/10.1007/978-3-319-67816-0_10 [90]. In dieser Arbeit wurden die Forschungsfragen 1, 2 und 3 erstmals exemplarisch beantwortet. Es wurde die Sinnhaftigkeit des Gesamtansatzes des Rahmenwerks exemplarisch dargestellt. Hierfür wurde ein Teil des Anforderungsmodells, eine erste Beschreibungssprache und erste für die beschriebenen Anforderungen geeignete Pseudonymisierungsverfahren umgesetzt und beschrieben.

Kasem-Madani S., Meier M. Utility Requirement Description for Utility-Preserving and Privacy-Respecting Data Pseudonymization. In: Gritzalis S., Weippl E.R., Kotsis G., Tjoa A.M., Khalil I. (eds) Trust, Privacy and Security in Digital Business. TrustBus 2020. Lecture Notes in Computer Science, vol 12395. Springer, Cham. 2020. https://doi.org/10.1007/978-3-030-58986-8_12 [89]. In dieser Arbeit wurde die Forschungsfrage 1 umfassend und die Fragen 2, 3 und 4 exemplarisch beantwortet. Es wurden die folgenden Komponenten des Rahmenwerks beschrieben: Das Anforderungsmodell, die Beschreibungssprache `Util` und die Pseudonymisierungsstruktur. Es wurden beispielhaft einzelne Übersetzungsregeln von den Anforderungen in geeignet parametrisierte Pseudonymisierungsverfahren beschrieben.

Kasem-Madani S., Malderle T., Boes F., Meier M. Privacy-Preserving Warning Management for an Identity Leakage Warning Network. In Proceedings of the European Interdisciplinary Cybersecurity Conference. Association for Computing Machinery, New York, NY, USA.

2020. https://doi.org/10.1145/3424954.3424955 [91]. In dieser Arbeit wurde die Forschungsfrage 2 exemplarisch für eine Nutzbarkeitsanforderung beantwortet. Es wurde ein Pseudonymisierungsverfahren vorgestellt, das in Kapitel 5.2.3 der vorliegenden Dissertation für die Umsetzung der Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Relation Element einer Menge angeführt wird. Im Kontext der Verwaltung der Warnung Betroffener von Identitätsdaten-Abfluss wird das Verfahren in Kapitel 7 der vorliegenden Arbeit in das Rahmenwerk eingebettet.

Weitere Veröffentlichungen Im Rahmen der Forschungsarbeiten zu dieser Dissertation sind die folgenden weiteren wissenschaftlichen Beiträge entstanden:

- Wendzel, S., Kasem-Madani. IoT Security: The Improvement-Decelerating 'Cycle of Blame' (Opinion Paper). 2016. https://doi.org/10.13052/popcas010. [149] In dieser kurzen Arbeit wurde der Frage nachgegangen, wie Unklarheiten in den Verantwortlichkeiten der Umsetzung von IT-Sicherheitsmechanismen zwischen verschiedenen Akteuren im Internet-of-Things Verbesserung der IT-Sicherheit behindern und wie Verbesserungen dennoch herbeigeführt werden können. Die grundlegende Idee, Nutzern ohne Expertenkenntnisse der IT-Sicherheit Hilfestellungen zu bieten, mit denen die IT-Sicherheit mittel- bis langfristig erhöht werden soll, wird in der vorliegenden Arbeit analog zur Verbesserung des Vertraulichkeitsschutzes personenbezogener Daten aufgegriffen.
- Christin D., Bub D. M., Moerov A., Kasem-Madani S. A Distributed Privacy-Preserving Mechanism for Mobile Urban Sensing Applications. 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). 2015. https://doi.org/10.1109/ISSNIP.2015.7106932. [41] In dieser Arbeit wird eine auf k-Anonymität basierende Anonymisierungstechnik für Trajektorien vorgestellt, mit der lokale Messwerte zwischen mobilen Geräten ohne genaue Kenntnis der Standorte der Geräte ausgetauscht werden können. Analog zu dieser Anonymisierungstechnik werden in der vorliegenden Arbeit Pseudonymisierungstechniken vorgestellt, die gezielt einzelne Nutzbarkeiten erhalten.
- Kasem-Madani, S., Michael M. Security and Privacy Policy Languages: A Survey, Categorization and Gap Identification. arXiv:1512.00201, 2015. https://arxiv.org/abs/1512.00201. [88] In dieser Übersichtsarbeit wurden maschinenlesbare Politiksprachen aus dem Kontext Sicherheit und Privacy identifiziert und kategorisiert. Es wurde untersucht, ob die Sprachen für die Beschreibung von Nutzbarkeits- und Vertraulichkeitsanforderungen an Pseudonymisierungen verwendet werden können. Diese Arbeit wird als Grundlage für die Erarbeitung des Standes der Wissenschaft der Beschreibungssprachen in Kapitel 4 verwendet.
- Kasem-Madani, S. A framework for encrypted computation on shared data. In: Meier, M., Reinhardt, D. & Wendzel, S. (Hrsg.), Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit. Bonn: Gesellschaft für Informatik e.V. (S. 191-196). 2016. https://dl.gi.de/bitstream/handle/20.500.12116/868/191.pdf [87] In dieser Arbeit wurde untersucht, wie durch Datentrennung und die Einführung unterschiedlicher Akteure auf homomorph verschlüsselten Daten operiert werden kann. Dies ist eine auf den Entwurf der Akteure des Rahmenwerks in Kapitel 1.2.1 und das Angreifermodell in Kapitel 1.2.2 vorbereitende Arbeit.

1.5 Grundlegende Begriffe

In dieser Dissertation wird ein grundlegendes Verständnis von Grundbegriffen und Verfahrenstypen aus den Themengebieten Datenschutz und Privacy zum einen und den Themengebieten Informationssicherheit und Privacy-Enhancing-Technologies zum anderen vorausgesetzt. Einige dieser Grundbegriffe werden in der Literatur nicht eindeutig mit derselben Bedeutung verwendet. In diesem Abschnitt werden daher die erforderlichen grundlegenden Begriffe eingeführt. Mögliche, in der Literatur häufig vorkommende Missverständnisse werden somit aufgelöst. Synonym verwendete Begriffe werden entsprechend gekennzeichnet.
Im Folgenden werden die Grundbegriffe nach den Themengebieten Datenschutz und Privacy bzw. Informationssicherheit und Privacy-Enhancing-Technologies kategorisiert. Für jedes Gebiet werden die Begriffe alphabetisch gelistet und beschrieben. Entstammen sie einer bestimmten Quelle, so wird diese angegeben. Definitionen ohne Angabe einer Quelle wurden für die vorliegende Dissertation erarbeitet.

1.5.1 Datenschutz und Privacy

Die Verarbeitung personenbezogener Daten geht einher mit Anforderungen, die zum Schutze der von der Verarbeitung Betroffenen formuliert werden. Für die Formulierung und Umsetzung dieser Anforderungen werden im Folgenden grundlegende Begriffe erläutert. Grundsätzlich wird bei der Datenverarbeitung zwischen Daten mit und ohne Personenbezug unterschieden. Im Sinne dieser Unterscheidung werden anonyme Daten definiert.

Definition 1: Anonyme Daten
Anonyme Daten sind Daten, die ohne Personenbezug erfasst wurden [128].

Anonyme Daten unterliegen nicht der Datenschutzgrundverordnung [151]. Ein Beispiel hierfür ist eine Zeitreihe der Außentemperatur eines bestimmten Ortes. Wenn diese Temperaturwerte nicht mit Individuen verknüpft werden können, sind die Daten nicht personenbeziehbar und damit anonym.

Definition 2: Anonymisierte Daten
Anonymisierte Daten sind ursprünglich personenbezogene Daten, die durch Anwendung einer oder mehrerer Anonymisierungstechniken in eine Form überführt wurden, die mit praktikablen Mitteln keinen Personenbezug mehr feststellen lässt [151].

Die praktikablen Mittel, die laut der Datenschutzgrundverordnung anzunehmen sind, umfassen alle rechtlich und technisch möglichen, realistisch eingesetzten Mittel zur Wiederherstellung eines Personenbezugs [55]. Ob Daten, die mit Personenbezug erhoben wurden, sicher anonymisiert werden können, wird stark angezweifelt und ist Gegenstand aktueller Forschung. Aktuell liegt die Vermutung nahe, dass eine Reidentifizierung aus anonymisierten Daten zwar erschwert und damit weniger wahrscheinlich ist, jedoch nicht vollständig ausgeschlossen werden kann. Dies wird zum

Beispiel von Rocher et al. beschrieben [127].

Definition 3: Betroffene
Betroffene im Sinne der Datenschutzgrundverordnung sind Personen, zu denen personenbezogene Daten verarbeitet werden [55].

Natürliche, identifizierte oder identifizierbare Personen, deren Daten verarbeitet werden, sind von dieser Verarbeitung Betroffene. Ihre Rechte werden durch verschiedene Gesetze geschützt. Hierzu zählt die europaweit gültige Datenschutzgrundverordnung [55].

Definition 4: Datenschutz
Datenschutz ist der Schutz personenbezogener Daten vor unerlaubter Verarbeitung [46].

Im Kontext dieser Arbeit dient der Datenschutz als Rahmen und Motivation für den Schutz der Vertraulichkeit personenbezogener Klartextdaten durch den Einsatz von Datenpseudonymisierung.

Definition 5: Datenschutzprinzipien
Die Datenschutzprinzipien nach Artikel 5 der Datenschutzgrundverordnung (Data Protection Principles) sind die Einhaltung der folgenden Prinzipien bei der Verarbeitung der Daten: die Rechtmäßigkeit, die Zweckbindung, die Datenminimierung, die Richtigkeit, die Speicherbegrenzung, die Integrität und Vertraulichkeit und die Rechenschaftspflicht [55].

Die Einhaltung der Datenschutzprinzipien soll die informationelle Selbstbestimmung [36] der Betroffenen und den Schutz der Vertraulichkeit personenbezogener Daten gewährleisten. Aus ihnen leitet sich die Anforderung zur Pseudonymisierung personenbezogener Daten ab.

Definition 6: Identifizierende Daten
Identifizierende Daten sind Daten, die eine direkte, eindeutige Identifizierung Betroffener ermöglichen.

Identifizierende Daten müssen zur Reidentifizierung Betroffener nicht mit anderen über das Individuum erfassten Daten kombiniert werden. Beispiele hierfür sind der volle Name und die deutsche steuerliche Identifikationsnummer[3]. Die Eindeutigkeit kann auch kontextabhängig sein. So kann zum Beispiel in einer Datensammlung von Namen der Beschäftigten eines Unternehmens ein Klarname eindeutig einem Individuum innerhalb dieses Unternehmens zugeordnet werden. Im Sprachraum, in dem das Unternehmen angesiedelt ist, kann derselbe Name durchaus mehrfach vorkommen. Von identifizierenden Daten[4] abzugrenzen sind quasi-identifizierende Daten[5]. Identifizierende und quasi-identifizierende Daten sind personenbezogene Daten.

[3] https://www.bzst.de/DE/Privatpersonen/SteuerlicheIdentifikationsnummer/steuerlicheidentifikationsnummer_node.html
[4] Siehe Definition 6
[5] Siehe Definition 11

Definition 7: Personenbezogene Daten
Personenbezogene Daten sind alle Informationen, die sich auf eine identifizierte oder identifizierbare lebende Person beziehen. Verschiedene Teilinformationen, die gemeinsam zur Identifizierung einer bestimmten Person führen können, stellen ebenfalls personenbezogene Daten dar [55].

Definition 8: Privacy
Privacy ist das Recht eines Einzelnen, seine persönlichen Angelegenheiten vertraulich zu halten [125].

Privacy ist als das Recht auf Achtung des Privat- und Familienlebens eines der Grundrechte der Charta der Grundrechte der Europäischen Union. Dort ist Privacy insbesondere in den Artikeln 7 und 8 in den Grundrechten Recht auf Schutz personenbezogener Daten und Recht auf Achtung des Privatlebens verankert [36].

Definition 9: Pseudonymisierung personenbezogener Klartextdaten (Datenpseudonymisierung)
Eine Pseudonymisierung personenbezogener Klartextdaten als Datenverarbeitungsprozess nach Artikel 4 der Datenschutzgrundverordnung ist die Verarbeitung personenbezogener Daten so, dass die pseudonymisierten Daten ohne Zuhilfenahme von Zusatzinformation nicht mehr einem spezifischen Datensubjekt zugeordnet werden können. Hierbei wird vorausgesetzt, dass die Zusatzinformation separat aufbewahrt wird und durch technische und organisatorische Maßnahmen so geschützt wird, dass personenbezogene Daten nicht einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können [55].

Durch die Datenpseudonymisierung werden personenbezogene Klartextdaten in eine Form überführt, die eine Reidentifizierung Betroffener durch Verkettung personenbezogener Information mit Identitäten im Vergleich zu Klartextdaten zumindest erschwert. Die Datenpseudonymisierung wird daher als risikomindernde Methode eingesetzt. Das Ergebnis der Durchführung von Datenpseudonymisierung ist eine Pseudonymisierung.

Definition 10: Pseudonymisierte Daten (Pseudonymisierung)
Eine Pseudonymisierung $P(D)$ einer Datensammlung D ist die Transformation von D in eine Repräsentation, die Definition 9 entspricht.

In der vorliegenden Arbeit wird die Pseudonymisierung einer Datensammlung D als $P(D)$ bezeichnet. Die Pseudonymisierung eines einzelnen Datums d aus D wird als $p(d)$ bezeichnet.

Definition 11: Quasi-identifizierende Daten
Quasi-identifizierende Daten sind personenbezogene Daten, die alleinstehend eine Reidentifizierung Betroffener nicht möglich machen. Werden sie jedoch mit anderen personenbezogenen Daten verkettet, so ist eine Reidentifizierung Betroffener möglich [146].

Beispiele für quasi-identifizierende Daten sind das Geburtsdatum, die aktuelle Meldeadresse, die Staatsangehörigkeit und medizinische Befunde. Besonders in älterer Literatur sind Ansätze zu

finden, in denen quasi-identifizierende von nicht-identifizierenden Daten getrennt werden sollen [113, 158]. Die als nicht-identifizierend eingestuften Daten wurden im Klartext verarbeitet. Die aktuelle Forschung zeigt jedoch, dass eine solche Differenzierung mit Risiken behaftet ist [132]. Anonymisierungen bzw. Pseudonymisierungen, die auf einer solchen Differenzierung aufbauten, konnten unter Nutzung der im Klartext verbliebenen Daten erfolgreich zur unintendierten Reidentifizierung Betroffener angegriffen werden [10, 132]. In der vorliegenden Arbeit wird daher angenommen, dass eine Pseudonymisierung keine Klartextdaten aus der Eingabedatenmenge enthält. Weitere Ausführungen hierzu befinden sich in Kapitel 4.4.

Definition 12: Reidentifizierung
Eine Reidentifizierung Betroffener mittels eines Datensatzes ist die eindeutige Verknüpfung eines Datensatzes mit der Identität der Betroffenen.

Die eindeutige Verknüpfung eines Datensatzes mit der Identität der einzelnen Betroffenen wird in der Literatur auch Verkettung genannt (siehe Hansen et al. zum komplementären Begriff der Nichtverkettbarkeit bzw. *Unlinkability* [74]). In dieser Arbeit bezeichnet die Verkettbarkeit eine Anforderungsklasse der Nutzbarkeiten. Diese ist in Kapitel 3 definiert. Die Ermöglichung der Verkettbarkeit ist bezüglich einer möglichen Nutzung zur unintendierten Reidentifizierung Betroffener risikobehaftet. Es gilt, dieses Risiko durch Ergreifen geeigneter Maßnahmen zu mindern.

Definition 13: Risikominderung
Mit Risikominderung werden alle Maßnahmen bezeichnet, die ergriffen werden, um das Risiko der Reidentifizierung Betroffener aus personenbezogenen Daten zu mindern.

Definition 14: Sensible Daten
Sensible Daten sind personenbezogene Daten, die sich auf besonders sensible Information über die Betroffenen beziehen [55].

Sensible Daten geben Informationen über für Betroffene kritische Information preis. Sie unterliegen besonderen Schutzbestimmungen [55]. Beispiele sind Information zur Ethnizität, der Einkommensklasse, der Religion und des Aufweisens bestimmter Erkrankungen.

1.5.2 Informationssicherheit und Privacy-Enhancing-Technologies

Für die Umsetzung von Datenschutz- und Privacy-Anforderungen an Systeme, die personenbezogene Daten verarbeiten, werden in dieser Arbeit Konzepte und Techniken aus der Informationssicherheit und den Privacy-Enhancing-Technologies verwendet und erweitert. Die für die Beschreibung dieser Umsetzung erforderlichen Begriff werden im Folgenden erläutert.

Definition 15: Anonymisierung
Anonymisierung ist die Anwendung von Techniken auf personenbezogenen Daten, die unter der Berücksichtigung realistisch zur Verfügung stehender Mittel eine Reidentifizierung Betroffener aus den Daten zumindest stark erschweren oder sogar gänzlich verhindern [151].

Ob Daten wirklich sicher anonymisiert werden können, ist eine offene Forschungsfrage. Es wird in der Forschung vermutet, dass sowohl Anonymisierungs- als auch Pseudonymisierungstechniken personenbezogene Daten allenfalls in eine Form überführen, mit der die Wiederherstellung des Personenbezugs der Daten mit einem erhöhten Aufwand verbunden ist [128].

Definition 16: Asymmetrische Verschlüsselungsverfahren
Asymmetrische Verschlüsselungsverfahren sind Verschlüsselungsverfahren, bei denen der Schlüssel aus einem Schlüsselpaar mit einem öffentlichen und einem privaten Schlüssel besteht. Der öffentliche Schlüssel wird für die Verschlüsselung der Klartextdaten verwendet. Der private Schlüssel wird für die Entschlüsselung der Chiffrate verwendet [86].

Asymmetrische Verschlüsselungsverfahren werden auch Public-Key-Verfahren genannt [86]. In dieser Arbeit werden sie mit unterschiedlichen Intentionen verwendet. Zum einen werden Eigenschaften bestimmter asymmetrischer Verfahren bei der Erstellung von nutzbarkeitserhaltenden Pseudonymisierungsverfahren[6] verwendet. Zum anderen werden probabilistische asymmetrische Verfahren genutzt, um die Verfügbarkeit der Nutzbarkeiten von Pseudonymisierungen zusätzlich einzugrenzen und zu kontrollieren[7].

Definition 17: Attributwert
Ein Attributwert ist der Wert eines Attributs eines Datensatzes. In semistrukturierten Daten[8] ist dies ein Spalteneintrag.

Personenbezogene Daten können als Datensammlung verarbeitet werden. Eine Datensammlung besteht aus Datensätzen. Ein Datensatz repräsentiert eine Zeile der Datensammlung. Eine Spalte einer Datensammlung beinhaltet alle Werte eines einzelnen Attributs. Ein Datensatz enthält Attributwerte in den Spalteneinträgen der Zeile des Datensatzes. Wenn in jedem Datensatz ein Wert für jedes Attribut der Datensammlung vorhanden ist, so ist die Datensammlung strukturiert. In diesem Falle induziert die Anordnung der Attribute und der Datensätze eine Relation. Werden nicht alle Werte in einem Datensatz belegt, so ist die Datensammlung semistrukturiert.
Bei der Verschlüsselung von Daten werden diese häufig in Datenblöcke eingeteilt. Der Verschlüsselungsalgorithmus wird dann auf den Datenblöcken ausgeführt.

Definition 18: Blockchiffre
Eine Blockchiffre ist ein Verschlüsselungsverfahren, bei dem die Operationen des Verfahrens wiederholt auf einem Datenblock ausgeführt werden.

Definition 19: Datenblock
Ein Datenblock ist eine Datenmenge fester Größe. Zur Erzeugung von Datenblöcken aus einer größeren Datenmenge wird diese in kleinere Mengen fester Größe (Blocklänge) unterteilt. Ist die

[6] Siehe Kapitel 5.
[7] Siehe Kapitel 6.
[8] Siehe Definition 50.

Größe der Datenmenge kein Vielfaches der Blocklänge, so wird die Differenz mit Dummy-Einträgen aufgefüllt.

Das Auffüllen eines Datenblocks mit Dummy-Einträgen ist eine Form des Paddings [118].

Definition 20: Datensammlung
Eine Datensammlung ist eine Menge von Datensätzen.

Häufig sind die Datensätze gleichförmig strukturiert. Datensammlungen enthalten die Attribute der enthaltenen Datensätze.

Definition 21: Datensatz
Ein Datensatz ist ein Tupel aus Daten, die wiederum Werte festgelegter Attribute sind. Für jedes der Attribute ist eine feste Position im Tupel bestimmt.

Ein Datensatz ist personenbezogen, wenn er Attributwerte zu bestimmten Individuen enthält.

Definition 22: Datenschutzfördernde Technik (DFT)
Siehe Definition 37 Privacy-Enhancing Technology (PET).

Definition 23: Datenschutzmechanismen
Siehe Definition 37 der Privacy-Enhancing Technology (PET).

Definition 24: Datum
Siehe Definition 17 des Attributwerts.

Definition 25: Chosen-Plaintext-Angriff
Ein Chosen-Plaintext-Angriff ist ein Angriff auf Kryptosysteme, bei dem dem Angreifer zu beliebigen Klartextdaten die entsprechenden Chiffrate vorliegen [17].

Chosen-Plaintext-Angriffe auf Pseudonymisierungsverfahren ermöglichen eine Aufdeckung der Klartextdaten ohne Brechen des Kryptosystems. Häufig stammen personenbezogene Daten aus einem vergleichsweise kleinen Suchraum, der durch eine Attribuierung des Datensatzes leicht zu ermitteln ist. Ein Beispiel sind weibliche Vornamen. Durch Zusatzinformation wie die Nationalität der Betroffenen kann der Suchraum weiter eingeschränkt werden. In der vorliegenden Arbeit wird diese Angriffsform daher besonders beachtet. In der Kryptographie existieren weitere Angriffsszenarien. Für eine Übersicht siehe zum Beispiel die Einträge zum Begriff *Attack* in [81].

Definition 26: Generalisierung
Generalisierung ist das Ersetzen kategorischer Werte durch verallgemeinernde, die Kategorie des Klartextdatums semantisch enthaltende Kategorien. Generalisierung ist eine Anonymisierungstechnik.

Definition 27: Hash-Funktion
Eine Hash-Funktion, auch Streuwertfunktion genannt, bildet eine große Eingabemenge auf eine kleinere Zielmenge ab.

Eine Hash-Funktion ist im Allgemeinen nicht injektiv. Die Eingabemenge kann Elemente unterschiedlicher Längen enthalten, die Elemente der Zielmenge haben dagegen meist eine feste Länge.

Definition 28: Hash-Funktion, kryptographische nach [53]
Eine kryptographische Hash-Funktion H ist eine nicht-injektive Funktion

$$H : X_1^* \longrightarrow X_2^k$$

für die gilt:

1. (Effizienz) Gegeben $x \in X_1^*$, dann ist $H(x)$ effizient berechenbar.
2. (Einwegeigenschaft, Preimage-Resistance) H ist eine Einwegfunktion: Bei gegebenem beliebigen $y = H(x)$ ist $x \in X_1^*$ mit $x = H^{-1}(y)$ nicht effizient zu berechnen.
3. (Schwache Kollisionsresistenz, Second-Preimage) Gegeben ein $x \in X_1^*$, dann ist es nicht effizient möglich, ein $x' \in X_1^*$ zu finden, so dass gilt: $x \neq x'$ und $H(x) = H(x')$.
4. (Starke Kollisionsresistenz) Es ist nicht effizient möglich, Paare $(x, x') \in X_1^* \times X_1^*$ zu finden, so dass gilt $x \neq x'$ und $H(x) = H(x')$.

Mit Hash-Funktionen als Pseudonymisierungsverfahren können Pseudonyme erzeugt werden, aus denen das zugrundeliegende Klartextdatum ohne erheblichen Mehraufwand nicht ermittelt werden kann.

Definition 29: Homomorphe Verschlüsselungsverfahren
Ein Verschlüsselungsverfahren E, das Klartextdaten $m \in M$ unter Nutzung eines Schlüssels k in Chiffrate $e_k(m) \in C$ überführt wird homomorph über einer Operation $\star$[1] genannt, wenn auf den Chiffraten eine Operation $*$ definiert ist und die folgende Gleichung gilt:

$$E(m_1) * E(m_2) = E(m_1) \star E(m_2) \forall m_1, m_2 \in M. \tag{1.1}$$

Mit auf homomorphen Verschlüsselungsverfahren als Pseudonymisierungsverfahren können Pseudonyme erzeugt werden, auf denen bestimmte Rechenoperationen ausgeführt werden können. Das Berechnungsergebnis der homomorph äquivalenten Berechnung auf den zugrundeliegenden Klartextdaten kann somit ohne Kenntnisnahme der Klartextdaten erzielt werden.

Definition 30: Informationsreduktion
Informationsreduktion ist die Überführung von Daten in eine Form, aus der nicht alle vor der Überführung aus den Daten schließbare Information abgeleitet werden kann.

Definition 31: Informationssicherheit (Security) nach [53]
Informationssicherheit (englisch: security) ist der Schutz der technischen Verarbeitung von Informationen und eine Eigenschaft eines funktionssicheren Systems. Sie soll verhindern, dass nicht-autorisierte Datenmanipulationen möglich sind oder die unautorisierte Preisgabe von Informationen stattfindet.

Definition 32: k–Anonymität
Eine Datensammlung erfüllt die k–Anonymität, wenn Information einer jeden in der Datensammlung repräsentierten Person nicht von mindestens $k-1$ anderen Personen unterschieden werden kann [131].

Die k-Anonymität erlaubt es, den Grad der Anonymität einer Datensammlung mathematisch abzuschätzen. Techniken zur Umsetzung von k-Anonymität reduzieren die Granularität der Repräsentation der Daten durch Ersetzen der Attributwerte durch zusammenfassende Werte. Hierbei entstehen Gruppierungen von Datensätzen, die eine eindeutige Zuordnung eines Datensatzes durch die Zuordnung der Attributwerte zu einem Betroffenen erschweren. Ziel ist hierbei die Verhinderung der Reidentifizierung Betroffener durch die Verarbeitung der Daten. Gleichzeitig wird eine Reduzierung der Genauigkeit der Daten hingenommen.

Definition 33: Kryptosystem
Ein Kryptosystem ist ein System, das aus einem Verschlüsselungsalgorithmus, einem Entschlüsselungsalgorithmus und einem Tripel wohldefinierter Mengen der Klartextdaten, der Chiffrate und der Schlüssel besteht [11].

Definition 34: Malleability (Formbarkeit)
Malleability ist die Eigenschaft von Verschlüsselungsverfahren, aus einem gegebenen Chiffrat y des Klartextdatums x ein gültiges Chiffrat $y' \neq y$ für x zu erzeugen [49].

Für die Ausnutzung von Malleability ist die Kenntnis des Klartextes nicht erforderlich. Beispiele für Verschlüsselungsverfahren, bei denen Malleability möglich ist, sind das Verfahren von Paillier [119] und das von Elgamal [54]. Soll die Integrität[9] von Chiffraten dieser Verfahren überprüft werden, muss dieser Umstand berücksichtigt werden.

Definition 35: Nutzbarkeitsanforderungen
Nutzbarkeitsanforderungen sind Anforderungen, die sich aus der intendierten Verarbeitung an die Daten ergeben.

Ist zum Beispiel vorgesehen, im Laufe der Datenverarbeitung die Daten auf Gleichheit mit anderen Daten zu überprüfen, so ist die Verkettbarkeit bezüglich der Relation Gleichheit eine Nutzbarkeitsanforderung an diese Daten.

[9] Siehe Definition 48 der Integrität als Schutzziel der Informationssicherheit.

Definition 36: Nutzbarkeitspolitik
Eine Nutzbarkeitspolitik ist eine genaue Beschreibung der Nutzbarkeiten, die ein Datensatz nach der Datenpseudonymisierung aufweisen soll. Zusätzlich werden die Vertraulichkeitsanforderungen angegeben, die bei der Verarbeitung der Pseudonymisierung erfüllt werden müssen.

In dieser Arbeit wird mit `Util` eine Beschreibungssprache erarbeitet, mit der Nutzbarkeitspolitiken mit Nutzbarkeitsanforderungen an Pseudonymisierungen maschinenlesbar formuliert werden können. Eine Nutzbarkeitspolitik dient auch der Dokumentation der Nutzbarkeiten, die in einer Pseudonymisierung aus den zugrundeliegenden Klartextdaten erhalten wurden.

Definition 37: Privacy-Enhancing-Technology (PET) nach Borking et al. [27, 28]
Privacy-Enhancing-Technologies (PET) sind technische Maßnahmen, die die Privacy schützen, indem sie personenbezogene Daten durch Löschung oder Informationsreduktion verändern, oder indem sie die unnötige Verarbeitung personenbezogener Daten verhindern. Gleichzeitig bleibt die Funktionalität des verarbeitenden Systems im Sinne der Anwendung erhalten.

Definition 38: Private-key-Verschlüsselungsverfahren
Siehe Definition 52 der symmetrischen Verschlüsselungsverfahren.

Definition 39: Probabilisierung
Die Probabilisierung ist die Überführung eines deterministischen kryptographischen Verfahrens in eine probabilistische Variante.

Mit einer geeigneten Probabilisierung können Verfahren gegen Angriffe gehärtet werden [47]. Ein Beispiel ist die Nutzung eines Salts[10] in Form eines einmaligen Zufallswertes als zweiten Parameter einer deterministischen kryptographischen Hash-Funktion. Das Ergebnis ist ein Verfahren, das für dasselbe Klartextdatum beim Einsatz unterschiedlicher Salt-Werte verschiedene Hash-Werte erzeugt. Unter bestimmten Umständen kann die Probabilisierung genutzt werden, um Wörterbuchangriffe zu erschweren.

Definition 40: Probabilistisches Verschlüsselungsverfahren
Ein probabilistisches Verschlüsselungsverfahren ist ein Verschlüsselungsverfahren mit der Eigenschaft, dass die mehrmalige Erzeugung eines Chiffrats durch Verschlüsselung desselben Klartextdatums unter Nutzung desselben Schlüssels zu unterschiedlichen Chiffraten führt.

Ein Sicherheitsvorteil von Verschlüsselungsverfahren mit dieser Eigenschaft ist, dass ein mehrfaches Vorkommen desselben Klartextdatums in den Chiffraten nicht ohne erheblichen Mehraufwand erkannt werden kann.

Definition 41: Pseudonym
Ein Pseudonym ist ein Datum, das ein personenbezogenes Klartextdatum innerhalb einer Pseudonymisierung ersetzt.

[10] Siehe Definition 46.

Sofern nicht anders gekennzeichnet, bezeichnet $p(d)$ in dieser Arbeit ein Pseudonym eines Klartextdatums d. Ist das Pseudonym ein einzelnes Utility-Tag[11] für eine Nutzbarkeit von d innerhalb einer Utility-Tag-Struktur[12] einer Pseudonymisierung, so wird es mit $u(d)$ gekennzeichnet. Soll die Zuordnung zur betreffenden Utility i kenntlich gemacht werden, so wird i als Index des Utility-Tags $u_i(d)$ gesetzt. In Kapitel 6 wird aufgezeigt, wie ein Pseudonym aus mehreren Utility-Tags bestehen kann.

Definition 42: Public-Key-Verschlüsselungsverfahren
Siehe Definition 16 der asymmetrischen Verschlüsselungsverfahren.

Definition 43: Randomisierung
Siehe Definition 39 der Probabilisierung.

Definition 44: Reidentifizierung
Reidentifizierung ist die eindeutige Verkettung der Identität Betroffener mit ihren personenbezogenen Daten.

Definition 45: Rerandomisierung
siehe Definition 34 der Malleability bzw. Formbarkeit.

Definition 46: Salt
Ein Salt ist ein Wert, der als Parameter in die Berechnung von kryptographischen Verfahren eingeht. Ein Salt wird verwendet, um das verwendete Verfahren in Abhängigkeit des Salts zu probabilisieren.

Definition 47: Schlüssel
Ein Schlüssel ist eine Information, die als Parameter eines Verschlüsselungsalgorithmus verwendet wird.

Definition 48: Schutzziele der Informationssicherheit nach Eckert [53]
Die Schutzziele der Informationssicherheit sind die Vertraulichkeit (englisch: Confidentiality), die Integrität (englisch: Integrity) und die Verfügbarkeit (englisch: Availability). Für das Schutzziel Vertraulichkeit wird gefordert, dass Daten zu jedem Zeitpunkt der Datenverarbeitung lediglich von autorisierten Benutzern gelesen bzw. modifiziert werden. Für das Schutzziel Integrität wird gefordert, dass Daten nicht unbemerkt und unautorisiert manipuliert werden. Jede Änderung der Daten muss nachvollzogen werden können. Für das Schutzziel Verfügbarkeit wird gefordert, dass Systemausfälle verhindert werden. Der Zugriff auf Daten muss entsprechend gültiger Vereinbarungen gewährleistet sein.

Die Schutzziele werden zum Erreichen und Einhalten der Informationssicherheit und damit zum Schutz der Daten vor beabsichtigten Angriffen von IT-Systemen definiert.

[11] Siehe Definition 54.
[12] Siehe Definition 55.

Definition 49: Semantische Sicherheit
Ein Verschlüsselungsverfahren ist semantisch sicher, wenn die Chiffrate des Verfahrens ohne erheblichen Mehraufwand keinerlei Information über die zugrundeliegenden Klartextdaten preisgibt [130].

Probabilistische Verschlüsselungsverfahren nach dem Stand der Wissenschaft sind semantisch sicher. Semantische Sicherheit ist äquivalent zur Ununterscheidbarkeit (indistinguishability, IND) [130].
In der vorliegenden Arbeit sind personenbezogene Daten Gegenstand der Betrachtung. Diese werden im großen Stil so strukturiert, dass ihre Position im Datensatz semantisch eine Rolle spielt. Häufig werden die Daten semistrukturiert verarbeitet.

Definition 50: Semistrukturierte Daten
Semistrukturierte Daten sind nach Attributen und Datensätzen strukturierte Datensammlungen, deren Datensätze nicht notwendigerweise Werte für alle in der Datensammlung definierten Attribute enthalten.

Ein Beispiel für semistrukturierte Daten sind Dateien im Comma-Separated-Value-Format [138].

Definition 51: Suppression
Suppression ist eine Anonymisierungstechnik, bei der identifizierende Daten aus dem zu anonymisierenden Datensatz gelöscht werden.

Wird die Suppression ausschließlich auf identifizierende Daten eines Datensatzes angewendet, so kann häufig aus den im Klartext vorliegenden quasi-identifizierenden Daten auf die Identität der Betroffenen geschlossen werden. Damit ist das Risiko der Reidentifizierbarkeit lediglich leicht reduziert.

Definition 52: Symmetrische Verschlüsselungsverfahren
Symmetrische Verschlüsselungsverfahren sind Verschlüsselungsverfahren, bei denen derselbe Schlüssel für die Ver- und Entschlüsselung verwendet wird. Dieser Schlüssel ist geheimzuhalten.

Definition 53: Technisch-organisatorische Schutzmaßnahme (TOM)
Eine technisch-organisatorische Schutzmaßnahme ist eine Maßnahme, die zur Erfüllung der Sicherheit personenbezogener Daten auf dem verarbeitenden System implementiert wird [46].

Definition 54: Utility-Tag
Ein Utility-Tag ist ein Pseudonym eines Klartextdatums, das eine dedizierte Nutzbarkeit (Utility) des Klartextdatums bereitstellt.

Eine Nutzbarkeit kann auf mehreren zu pseudonymisierenden Klartextdaten definiert werden. Das Pseudonym der Nutzbarkeit ist dann eine sogenannte Utility-Tag-Struktur, die alle Utility-Tags der Klartextdaten enthält, für die diese Nutzbarkeit gefordert ist.

Definition 55: Utility-Tag-Struktur
Eine Utility-Tag-Struktur ist ein Pseudonym einer Nutzbarkeit, das eine Menge von Utility-Tags umfasst. Jedes der Utility-Tags entspricht einem der von der Nutzbarkeit adressierten Klartextdaten.

Utility-Tags werden durch die Anwendung von Pseudonymisierungsverfahren auf Klartextdaten erzeugt. Dies können eigenschaftenerhaltende Verschlüsselungsverfahren sein.

Definition 56: Verschlüsselungsverfahren
siehe Definition 33 des Kryptosystems.

Datenpseudonymisierung dient in dieser Arbeit als Maßnahme, den Schutz der Vertraulichkeit personenbezogener Daten sicherzustellen und das Risiko der Reidentifizierung Betroffener durch eine Verarbeitung dieser Daten zu mindern.

Definition 57: Vertraulichkeitsanforderungen
Vertraulichkeitsanforderungen sind Anforderungen an eine Pseudonymisierung, mit der die Vertraulichkeit der zugrundeliegenden Klartextdaten und die Identität der von der Datenverarbeitung Betroffenen soweit wie möglich gewährleistet werden soll.

Definition 58: Wörterbuchangriff
Ein Wörterbuchangriff nach [18] ist ein Ansatz der Kryptoanalyse in der der Angreifer Klartext-Chiffrat-Paare nach den Chiffraten sortiert erzeugt und speichert. Die Klartexte werden aus häufig vorkommenden Daten gewählt. Werden die Chiffrate durch Verschlüsselungsverfahren erzeugt, so werden die Klartext-Chiffrat-Paare für jeden vorliegenden Schlüsselkandidaten einzeln generiert. Liegt nun ein verschlüsselter Klartext vor, so durchsucht der Angreifer die ihm vorliegenden Klartext-Chiffrat-Paare nach dem entsprechenden Chiffrat und ermittelt so den zugrundeliegenden Klartext und den verwendeten Schlüssel.

Wörterbuchangriffe sind eine Möglichkeit, die Klartexte von Hash-Werten und Chiffraten von deterministischen asymmetrischen Verschlüsselungsverfahren zu ermitteln. Wörterbuchangriffe gehören zu den Chosen-Plaintext-Angriffen. Sie stellen eine Herausforderung an den Entwurf sicherer Pseudonymisierungsverfahren dar.

2 Überblick über den Ansatz

In diesem Kapitel wird der Gesamtansatz der Arbeit zur nutzbarkeitserhaltenden Datenpseudonymisierung beschrieben. Der Gesamtansatz umfasst die grundlegenden Komponenten Pseudonymisierung mit Utility-Tags, Anforderungen an Pseudonyme, Beschreibungssprache `Util` und Übersetzungsregeln. Ziel dieses Kapitels ist es, einen Überblick der Komponenten des Rahmenwerks zu liefern. Diese werden daher im Folgenden kompakt dargestellt.
Um in einer Pseudonymisierung mehrere Nutzbarkeiten einzelner Klartextdaten nach Bedarf und automatisiert bereitstellen zu können, wird eine flexible Struktur zur Generierung und Haltung der Pseudonymisierung benötigt. Zunächst soll daher ein Überblick über die in dieser Arbeit in Kapitel 6 entwickelte Struktur für eine nutzbarkeitserhaltenden Pseudonymisierung, die sog. Pseudonymisierung mit Utility-Tags, gegeben werden.
Um nutzbarkeitserhaltende Pseudonyme erstellen zu können, müssen die Anforderungen an die Pseudonymisierung bestimmter Klartextdaten formuliert werden können. Hierzu werden in Kapitel 3 grundlegende Anforderungsklassen der Nutzbarkeits- und der Vertraulichkeitsanforderungen hergeleitet und beschrieben.
Für eine automatisierte Erstellung von Pseudonymisierungen müssen die Anforderungen an eine solche maschinenlesbar formuliert werden können. Hierfür wird in dieser Arbeit die Beschreibungssprache `Util` entwickelt und in Kapitel 4 beschrieben.
Um aus einer maschinenlesbaren Beschreibung der Anforderungen an eine Pseudonymisierung von Klartextdaten automatisiert eine Pseudonymisierung mit Utility-Tags ableiten zu können, werden geeignete Übersetzungsregeln benötigt. In Kapitel 6 dieser Arbeit werden daher für eine Reihe von Nutzbarkeits- und Vertraulichkeitsanforderungen Regeln beschrieben, mit denen ausgehend von einer Anforderungsbeschreibung in `Util` eine diesen Anforderungen genügende Pseudonymisierung mit Utility-Tags hergeleitet werden kann. Das erarbeitete Rahmenwerk ist in Abbildung 3 skizziert.

2.1 Anforderungen an Pseudonyme

Ziel der nutzbarkeitserhaltenden Datenpseudonymisierung ist die Bereitstellung personenbezogener Daten in einer Form, die zum einen dediziert bestimmte Nutzbarkeiten der zugrundeliegenden personenbezogenen Klartextdaten aufrechterhält, zum anderen dennoch die Vertraulichkeit zugrundeliegender personenbezogener Daten schützt. Hierzu soll eine nutzbarkeitserhaltende Pseudonymisierung bestimmten Anforderungen genügen. Diese Anforderungen können im Wesentlichen in zwei Anforderungsklassen formuliert werden: Vertraulichkeitsanforderungen und Nutzbarkeitsanforderungen.

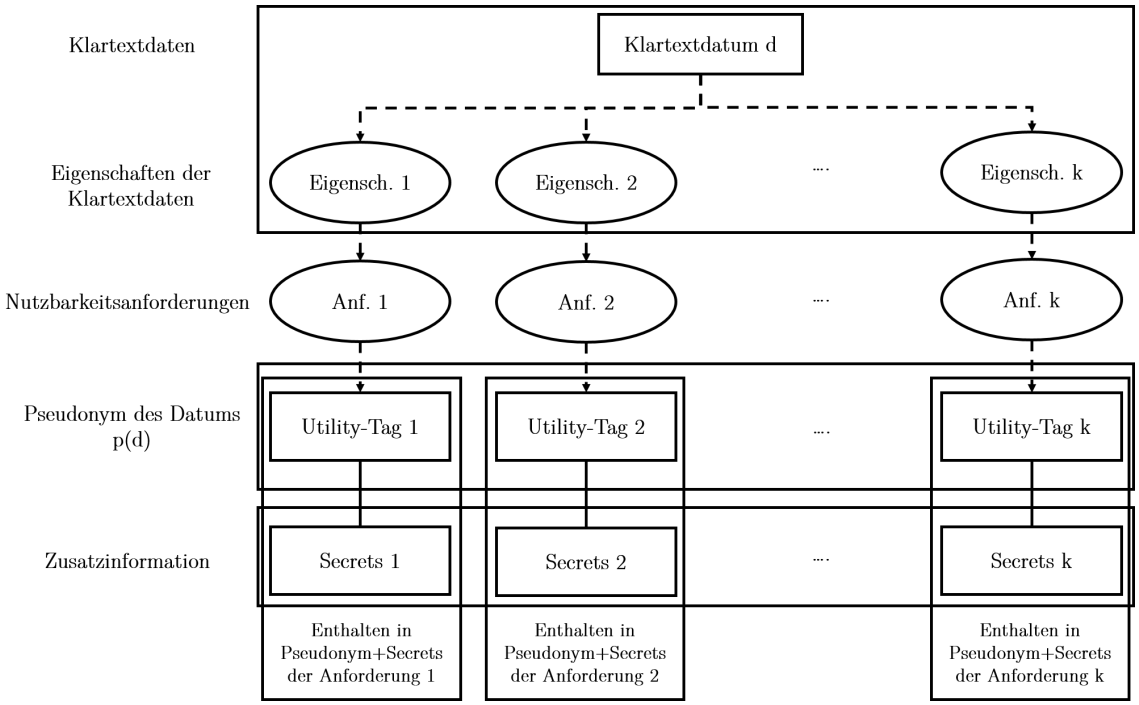

Abbildung 3: *Überblick der Erstellung von Pseudonymisierungen mit Utility-Tags nach dem in dieser Arbeit erarbeiteten Ansatz.*

Vertraulichkeitsanforderungen sind Anforderungen an Pseudonymisierungen, deren Umsetzung den Schutz der Vertraulichkeit der zugrundeliegenden Daten zum Ziel hat. Sie werden ausdrücklich nicht definiert, um aus den resultierenden Eigenschaften der abgeleiteten Pseudonymisierung bestimmte Nutzbarkeiten ableiten zu können. Vielmehr folgt aus der Umsetzung der Anforderungen typischerweise eine Einschränkung der Nutzbarkeiten einer Pseudonymisierung. Je nach Art der Einschränkung können verschiedene Vertraulichkeitsanforderungen formuliert werden. Diese werden in Kapitel 3.3 systematisiert.
Nutzbarkeitsanforderungen beschreiben Anforderungen, die aus sog. Nutzbarkeiten, also Charakteristiken der den Pseudonymen zugrundeliegenden Klartextdaten stammen. Wird eine Nutzbarkeit eines Klartextdatums zur sinnvollen Berechnung benötigt, so muss sie in der Pseudonymisierung so repräsentiert sein, dass geeignete Berechnungen auf der Pseudonymisierung durchgeführt werden können. Die Berechnungen auf der Pseudonymisierung sollen ohne die Erfordernis des direkten Zugriffs auf die zugrundeliegenden Klartextdaten möglich sein. Ausgehend von gegebenen Klartextdaten können verschiedene Nutzbarkeitsanforderungen formuliert werden. Die Systematisierung dieser Anforderungen wird in Kapitel 3.2 beschrieben.
Nutzbarkeitsanforderungen werden mit Vertraulichkeitsanforderungen kombiniert. Ziel des Kombinierens ist das Erreichen eines für die geplanten Berechnungen maximalen Schutzes der Vertraulichkeit der Klartextdaten bei gleichzeitiger Verfügbarkeit der Information, deren Ermittlung durch die geplante Berechnung intendiert ist.

2.2 Beschreibungssprache

In der vorliegenden Arbeit soll u.a. aufgezeigt werden, dass die in Kapitel 2.1 und später in Kapitel 3 angeführten Anforderungen an Pseudonyme für eine automatisierte Aufbereitung von flexibel konfigurierbaren nutzbarkeitserhaltenden Datenpseudonymisierungen verwendet werden können. Eine automatisierte Aufbereitung von nutzbarkeitserhaltenden Pseudonymen erfordert die maschinenlesbare Angabe dieser Anforderungen. Hierfür wurde in dieser Arbeit die Beschreibungssprache `Util` entwickelt. Sie ermöglicht die nutzbarkeitsorientierte Definition der Anforderungen an eine Pseudonymisierung einer bestimmten Datensammlung und ist in `XML` deklariert. Es werden zunächst die in der zu erstellenden Pseudonymisierung geplanten Nutzbarkeiten deklariert. Für jede der Nutzbarkeiten werden alle Klartextdaten aus der Datensammlung referenziert, die gemeinsam zur Ausführung der Nutzbarkeit benötigt werden. Für jede einzelne Nutzbarkeit werden Vertraulichkeitsanforderungen formuliert, die zu einer Beschränkung der Nutzbarkeit der Pseudonyme im Sinne geltender Datenschutzbestimmungen führen sollen. Diese Beschränkung soll zu einem über die Datenpseudonymisierung hinausgehenden mittelbaren Schutz der zugrundeliegenden Klartextdaten führen. Ergebnis der Deklarierung in `Util` für eine Datenmenge und eine geplante Anwendung der zu pseudonymisierenden Daten ist eine maschinenlesbare Nutzbarkeitspolitik.
Die Beschreibungssprache `Util` dient also als Mensch-Maschine-Schnittstelle zur Beschreibung der Anforderungen, die eine zu erstellende Pseudonymisierung erfüllen soll. Um diese Schnittstelle auch für Nichtexperten im Bereich der Privacy-Enhancing-Technologies (PET) zugänglich zu machen, wurde `Util` so entworfen, dass die Beschreibung der Nutzbarkeitsanforderungen und der Vertraulichkeitsanforderungen mit minimalen Kenntnissen der PET auskommt. Insbesondere werden Kenntnisse in der Angewandten Kryptographie, wie sie für die Erstellung einer sinnvollen Pseudonymisierung erforderlich sind, nicht für die Angabe der Anforderungen in `Util` benötigt. Die Beachtung dieser Umstände soll zu einer Erleichterung der Erstellung von Pseudonymisierungen durch Nichtexperten führen. Insgesamt soll mit dieser Arbeit ein Beitrag zur Verbreitung und Nutzbarkeit von Pseudonymisierungsverfahren und PET im Allgemeinen geleistet werden.
Zusätzlich zu der Verwendung von in `Util` deklarierten Anforderungen zur automatisierten Ableitung von Pseudonymisierungen wird eine Nutzbarkeitspolitik auch als Dokumentation der aus den zugrundeliegenden Klartextdaten in einer Pseudonymisierung zur Verfügung gestellten Information verwendet. Weiterhin ist eine Nutzbarkeitspolitik Teil der Dokumentation der in einem System eingesetzten technisch-organisatorischen Maßnahmen.

2.3 Pseudonymisierung mit Utility-Tags

Pseudonymisierung mit Utility-Tags wurden in dieser Arbeit als flexible Datenstruktur entwickelt, um die in einer `Util`-Nutzbarkeitspolitik formulierten Anforderungen umsetzen zu können. Diese Datenstruktur ermöglicht eine automatisierte Generierung und Verarbeitung von Pseudonymisierungen mit unterschiedlichen Nutzbarkeiten. Somit wird insbesondere der Variabilität in der Umsetzung durch die verschiedenen nutzbarkeitserhaltenden Pseudonymisierungsverfahren begegnet.
In dieser Datenstruktur ist eine Pseudonymisierung einer Datenmenge eine geordnete Menge

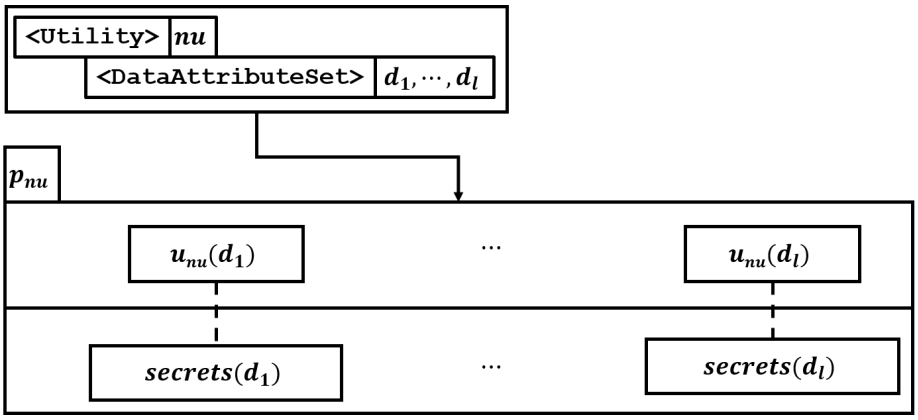

Abbildung 4: *Datenstruktur einer Pseudonymisierung mit Pseudonymen mit Utility-Tags einer Nutzbarkeit nu und der zugehörigen Secrets-Struktur.*

von Pseudonymen. Jedes Pseudonym entspricht einer Sequenz sog. Utility-Tags. Ein Utility-Tag ist hierbei die Ausprägung einer Nutzbarkeit eines einzelnen Klartextdatums. Für eine in einer Nutzbarkeitspolitik deklarierte Nutzbarkeitsanforderung wird für jedes innerhalb dieser Anforderung adressierte Klartextdatum im entsprechenden Pseudonym ein Utility-Tag erstellt, das für diese Nutzbarkeit zur Ausführungszeit ausgewertet wird. Ist für die Auswertung der Nutzbarkeit sogenannte Zusatzinformation erforderlich, so wird diese in einer separaten, mit dem Pseudonym verknüpften sogenannten Secrets-Struktur angelegt. Dies erlaubt die Separierung von Pseudonymisierung und Zusatzinformation und stärkt somit die Vertraulichkeitserhaltung der Pseudonymisierung. Zum Beispiel kann so ein durch Entschlüsselung aufdeckbares Pseudonym als Utility Tag aufgefasst werden. Der Entschlüsselungsschlüssel wird in der zu diesem Utility-Tag zugehörigen Secrets-Struktur abgelegt.
Eine grafische Aufbereitung der Datenstruktur kann Abbildung 4 entnommen werden. In Kapitel 6 wird die Datenstruktur genauer beschrieben.

2.4 Übersetzungsregeln

Um eine in `Util` formulierte Nutzbarkeitspolitik zur automatisierten Ableitung einer passgenauen Pseudonymisierung nutzen zu können, ist die Ausführung von geeigneten Pseudonymisierungsverfahren auf den Klartextdaten erforderlich. Diese Verfahren müssen geeignet konfiguriert und parametrisiert werden. Um dies zu erreichen, werden in dieser Arbeit sog. Übersetzungsregeln vorgestellt. Diese Regeln erlauben eine automatisierte, flexible Ableitung von geeignet konfigurierten und parametrisierten nutzbarkeitserhaltenden Pseudonymisierungsverfahren. In Kapitel 6 findet sich eine Systematisierung und detaillierte Beschreibung der erarbeiteten Übersetzungsregeln.

2.5 Anwendungsbeispiele

Das in dieser Dissertation erarbeitete Rahmenwerk wurde mit dem Ziel entwickelt, nutzbarkeitserhaltende Pseudonymisierungsverfahren praktikabler zu machen. Um die Praktikabilität des Ansatzes und die grundlegende Eignung für den Einsatz in der Praxis nachzuweisen, wurden zwei Anwendungsbeispiele erarbeitet. Diese Anwendungsbeispiele umfassen eine Umfrageplattform und eine Abfragemöglichkeit geleakter Identitätsdaten. Sie werden in Kapitel 7 detailliert beschrieben.

3 Anforderungen an nutzbarkeitserhaltende Pseudonymisierungen

Um personenbezogene Daten risikomindernd verarbeiten zu können, sollen die Daten möglichst zeitnah nach ihrer Erfassung pseudonymisiert werden [55, 136]. Damit die pseudonymisierten Daten für den Anwendungsfall nutzbar sind, müssen sie bestimmte Eigenschaften aufweisen. In diesem Kapitel wird ein Anforderungsmodell erarbeitet, dessen Anforderungen Pseudonymisierungen zur Erfüllung des Zwecks der Datenverarbeitung und dem Schutz der Vertraulichkeit zugrundeliegender personenbezogener Information genügen sollen.
Zur Erfüllung des Zwecks der Datenverarbeitung müssen die Pseudonymisierungen die Nutzbarkeit der zugrundeliegenden Klartextdaten in gewissem Umfang erhalten bleiben. Die Pseudonymisierungen sollen gezielt so erstellt werden, dass sie bestimmten *Nutzbarkeitsanforderungen* genügen. Die Pseudonymisierungen und die zugrundeliegenden, in den Pseudonymisierungen zu erhaltenden Eigenschaften sind im Allgemeinen als personenbezogen zu betrachten[1]. Es ist daher davon auszugehen, dass sie ohne zusätzliche technisch-organisatorische Maßnahmen bis zu einem gewissen Umfang zur Reidentifizierung Betroffener bzw. Kenntnisnahme sensibler Information über Betroffene genutzt werden können. Daher unterliegen diese weiterhin dem besonderen Schutz der geltenden Datenschutzrichtlinien [55][136]. Die Umsetzung dieser Richtlinien soll die Vertraulichkeit der den Pseudonymisierungen zugrundeliegenden Klartextdaten sicherstellen. Auch soll das Risiko der Reidentifizierung Betroffener unter Nutzung von Klartextdaten und Information, die aus den Klartextdaten in die Pseudonymisierung eingeflossen ist reduziert werden. Die damit verbundenen Anforderungen werden als *Vertraulichkeitsanforderungen* formuliert.
Ziel der Erarbeitung von Anforderungsklassen für Pseudonymisierungen ist die Bereitstellung eines Rahmenwerks zur Formulierung von Nutzbarkeitsanforderungen mit Vertraulichkeitsanforderungen. Dieses Rahmenwerk soll mit Expertenkenntnissen auf dem Anwendungsgebiet, jedoch ohne spezifische Kenntnisse über die Umsetzung der Nutzbarkeitsanforderungen unter Verwendung von PET angewendet werden können. Zur Formulierung der Vertraulichkeitsanforderungen sollten ebenfalls keine Expertenkenntnisse in PET erforderlich sein. Vielmehr ist das Ziel, die Vertraulichkeitsanforderungen so zu formulieren, dass sie die Angabe PET-relevanter Information in möglichst einfacher, aber dennoch ausreichender Form erlaubt. Innerhalb des Rahmenwerks sollen möglichst umfassende Nutzbarkeitsanforderungen formuliert werden können. Die Nutzbarkeitsanforderungen sollen nach Möglichkeit anwendungsfallunabhängig formuliert werden.

[1] Siehe z.B. die Ausführungen zum entsprechenden Gerichtsurteil [144].

Sehr anwendungsspezifische Benennungen von Anforderungen, die auf der Ebene der Datenverarbeitung die Ausführung derselben Operationen zur Folge haben, sollen daher vermieden werden. Das Anforderungsmodell bewirkt somit eine Systematisierung der Anforderungen an Pseudonymisierungen. Diese erleichtert eine gezielte Auswahl und Umsetzung maßgeschneiderter Pseudonymisierungen. Auch soll sie verhindern, dass dieselben Anforderungen in unterschiedlichen Anwendungen so kontextabhängig formuliert werden, dass geeignete Pseudonymisierungsverfahren in den unterschiedlichen Veröffentlichungsformaten wiederholt als neu vorgestellt werden. Wendzel et al. haben mit der Systematisierung und Kategorisierung von Netzwerk-Covert-Channels [150] bereits einen strukturell ähnlichen Ansatz im Kontext des Information-Hidings vorgestellt.

3.1 Stand der Wissenschaft

In einer Vielzahl von wissenschaftlichen Arbeiten werden Anforderungen an zu pseudonymisierende textuelle Daten formuliert. Diese werden im Folgenden beschrieben.

3.1.1 Direkte Formulierung von Nutzbarkeits- und Vertraulichkeitsanforderungen

Es werden die folgenden drei Anforderungen implizit oder explizit formuliert: (1) Die zumindest eingeschränkte Reidentifizierbarkeit Betroffener durch Abbildung von Pseudonymen auf die zugrundeliegenden Klartextdaten, (2) die Verkettbarkeit von personenbezogenen Daten mit den identifizierenden Daten der Betroffenen und (3) die Verkettbarkeit gleicher Klartextdaten innerhalb einer Datensammlung. Beispiele, in denen zumindest eine dieser Anforderungen formuliert und der Anforderung genügende Pseudonymisierungsverfahren beschrieben werden sind in einer Vielzahl von Anwendungsfällen zu finden. Einige werden im Folgenden exemplarisch beschrieben. Heurix et al. [76] formulieren Anforderungen an die Pseudonymisierung von textuellen Daten als Erhaltung eines Klartextes im Ganzen, Löschen des Klartextes oder Ersetzen des Klartextes durch einen Platzhalter. In der Arbeit von Bissmeyer et al. [22] werden aufdeckbare Pseudonyme zur bedingten Reidentifizierung bestimmter Fahrzeuge beschrieben. Neubauer et al. beschreiben in [114] die Pseudonymisierung von elektronischen Gesundheitsdaten (engl. electronic health records), die von autorisierten Subjekten bestimmter Rollen aufgedeckt werden können. So kann die Reidentifizierung bestimmter Patienten auf Subjekte eingeschränkt werden, die im System eine bestimmte Rolle einnehmen. Ein Beispiel einer solchen Rolle ist der aktuell behandelnde Arzt. In der Arbeit [115] wird die Aufdeckbarkeit der Pseudonyme für Primär- und Sekundärzwecke beschrieben.
In den Arbeiten von Flegel und Biskup [61] und von Flegel und Meier [63] werden Anforderungen an Pseudonyme im Kontext der Logdatenanalyse im verteilten Monitoring und in der Einbruchsdetektion formuliert. Diese sind die Verkettbarkeit von Pseudonymen bzgl. der Gleichheit der zugrundeliegenden Klartextdaten und die bedingte Aufdeckbarkeit. Diese beiden Anforderungen umfassen im Wesentlichen die Operationen, die im Rahmen der Analyse der Logdateien erforderlich sind. Die Verkettbarkeit bzgl. Gleichheit erlaubt das Erkennen von kritischen Aktivitäten einer

einzelnen Entität. Die eingeschränkte Aufdeckbarkeit erlaubt eine Aufdeckung z.B. der IP-Adressen von Entitäten mit verdächtigen Aktivitäten.
In den Arbeiten von Biskup, Flegel, Meier und weiteren [60, 59, 61, 62] im Bereich der Logdatenanalyse werden Anforderungen an die Verkettbarkeit und Aufdeckbarkeit von Pseudonymen erstmals explizit als sogenannte Verfügbarkeitsanforderungen formuliert. Hintergrund ist deren Arbeit zur Pseudonymisierung von personenbezogenen Daten in Logdateien. Diese werden im Rahmen des Monitorings in Form von aufgezeichneten Aktivitäten einzelner Nutzer unter Erfassung der zu dem Zeitpunkt vergebenen IP-Adresse analysiert. Hierbei ist z.B. von Interesse, ob eine IP-Adresse mehrfach durch bestimmte Aktivitäten aufgefallen ist. Um dies auf pseudonymisierten Daten erkennen zu können, werden die IP-Adressen als personenbezogene Daten so pseudonymisiert, dass sie verkettbar bezüglich der Relation Gleichheit sind. Die zeitliche Abfolge als Erhaltung einer Ordnungsrelation über die Zeitstempel wird implizit über die Erhaltung der Reihenfolge der aufgezeichneten Aktivitäten beibehalten. Wird im Rahmen der Analyse der Logdaten potenziell maliziöse Aktivität erkannt, so soll der Klartext der zugehörigen IP-Adresse aufgedeckt werden. Erstmals formulieren die Autoren mit der Rollen- und Zweckbindung auch explizite Vertraulichkeitsanforderungen an Pseudonyme.
Die Beschreibung der Anforderungsklassen in der vorliegende Dissertation baut auf den oben genannten Arbeiten von Flegel et al. auf, indem es die Anforderungsklassen erweitert. Die Nutzbarkeitsanforderungen werden um Klassen erweitert, die für die Datenanalyse erforderlich sind. Die Vertraulichkeitsanforderungen werden so erweitert, dass zusätzlich zur Bindung der Verfügbarkeit der Pseudonyme an Rollen und Zwecke die mittlerweile etablierten Datenschutzprinzipien adressiert werden können.
Hansen, Jensen und Rost beschreiben in [74] Schutzziele für die Entwicklung von PET und den Konflikt zwischen Verfügbarkeit und Vertraulichkeit personenbezogener Daten. Diese sind zum einen die drei Schutzziele der IT-Sicherheit: Vertraulichkeit, Integrität und Verfügbarkeit. Hinzu kommen die Nichtverkettbarkeit, die Transparenz und die Intervenierbarkeit. Die Nichtverkettbarkeit wird definiert als Verhinderung der Herstellung einer Verbindung zwischen personenbezogenen Daten einer Betroffenen über unterschiedliche Kontexte hinweg. Transparenz ist die Nachvollziehbarkeit und Rekonstruierbarkeit der auf personenbezogenen Daten durchgeführten Verarbeitung. Intervenierbarkeit ist die Möglichkeit, in die Verarbeitung personenbezogener Daten korrigierend oder löschend eingreifen zu können. Eine dort implizit beschriebene Nutzbarkeitsanforderung ist die Verkettung von Identitäten mit personenbezogenen Daten. Die Schutzziele werden in der vorliegenden Arbeit in den Vertraulichkeitsanforderungen gespiegelt. Im Sinne der mehrschichtigen Sicherheit werden hierbei die Schutzziele der IT-Sicherheit Integrität und Verfügbarkeit als durch das verarbeitende System umzusetzende Ziele betrachtet. Die Vertraulichkeit, die Nichtverkettbarkeit, die Transparenz und die Intervenierbarkeit werden durch die Umsetzung der Pseudonymisierung unter Berücksichtigung der in Abschnitt 3.3 beschriebenen Vertraulichkeitsanforderungen umgesetzt. Hierbei ergibt sich die Nichtverkettbarkeit durch die Einschränkung bestimmter Verkettbarkeiten auf einzelne Daten unter Berücksichtigung von Rollen- oder Zweckbindung. Die Transparenz folgt aus der Nachvollziehbarkeit der Datenverarbeitung durch die Formulierung der umzusetzenden Anforderungen und den daraus folglich möglichen Datenverarbeitungen. Die Intervenierbarkeit folgt aus der Tatsache, dass jedes pseudonymisierte Klartextdatum in einer Pseudonymisierung

identifiziert und bei Bedarf aus dieser entfernt werden kann. Die von Hansen et al. beschriebene Verkettbarkeit wird in der Nutzbarkeitsanforderung der Verkettbarkeit aufgegriffen.
Zimmer et al. [159] definieren Anforderungen an zu pseudonymisierende Event-Log-Daten im Rahmen der Analyse der Daten für die Reaktion auf IT-Sicherheitsvorfälle. Die Event-Log-Daten werden von einem Sicherheitsmonitoring-System analysiert. Diese Anforderungen sind die Reidentifizierung bei Erkennung der Erfordernis einer Reaktion, die eingeschränkte Verkettbarkeit von Pseudonymen, die Verhinderung der Erkennung mehrmaliger Nutzung eines Pseudonyms und die Verhinderung der Verkettbarkeit mehrerer Pseudonyme. Ziel ist die eingeschränkte Aufdeckung von IP-Adressen bei Bedarf. Gleichzeitig soll verhindert werden, dass das Sicherheitsmonitoring-System die pseudonymisierten Daten zur Überwachung von Nutzeraktivitäten bei gleichzeitiger Reidentifizierung dieser ohne vorliegenden Sicherheitsvorfall nutzen kann. Im Gegensatz zur vorliegenden Arbeit ist diese Arbeit stark anwendungsspezifisch. Auch ist eine separate Ableitung von Nutzbarkeits- bzw. Vertraulichkeitsanforderung nicht direkt möglich. Daher kann die Arbeit nicht ohne weitreichende Modifikationen auf andere Anwendungsfälle übertragen werden.
Anwendungsfälle, in denen bestimmt werden soll, ob eine gegebene Menge von personenbezogenen Daten in einer bei Dritten vorliegenden Menge enthalten sind, sind wohlbekannt. Ein Beispiel ist die Abfrage von personenbezogenen Daten bei Drittparteien einer Branche zur Warnung vor Betrug. Hier verhindern rechtliche Rahmenbedingungen den Austausch und Abgleich der personenbezogenen Daten im Klartext. Daher werden Nutzbarkeitsanforderungen des Findens eines Elements in einer Menge ohne Kenntnisnahme der zugrundeliegenden Klartextdaten formuliert. Arp et al. haben diese Anforderung im Rahmen der Erfordernis der Anwendbarkeit bestimmter Filter formuliert [7]. Eine eindeutigere, direkte Formulierung findet sich in der vorliegenden Dissertation in Kapitel 7.2. Diese wurde bereits in der Arbeit [91] veröffentlicht.
Die Anforderung der Berechnung festgelegter Funktionen auf Chiffraten ist ebenfalls wohlbekannt. Ein Beispiel hierfür ist die Anforderung, regelungstechnische Berechnungen auf homomorph verschlüsselten Daten durchführen zu können [135].

3.1.2 Indirekte Anforderungsformulierung durch Angabe der Verfahren

Anforderungen können durch die Benennung des einzusetzenden Pseudonymisierungsverfahrens beschrieben werden. Hierzu zählt zum Beispiel die Arbeit von Slagell et al. [140]. Die Verfahren, die als Anforderung im dort beschriebenen Rahmenwerk angegeben werden können, umfassen das Schwärzen, die zufällige Permutierung, die Kürzung und die Präfix-Erhaltung. Aus den Eigenschaften der so erzeugten Pseudonyme kann auf die Nutzbarkeit dieser geschlossen werden. Jedoch ist der Umfang der ermöglichten Nutzbarkeiten für Nichtexperten nur schwer nachzuvollziehen. Durch die eingeschränkte Auswahl an Nutzbarkeiten kann die Pseudonymisierung im Allgemeinen nicht maßgeschneidert für die Anforderungen erstellt werden. Vielmehr ist die Erhaltung zumindest von Teilen der Klartextdaten das eingesetzte Mittel zur Erhaltung von Nutzbarkeit. Der Schutz von Vertraulichkeit erfolgt durch teilweise Informationsreduktion. Der die Pseudonymisierung erstellende Nutzer muss in der Lage sein, ohne vorgegebene Hilfsmittel aus den anzugebenden Techniken die möglichen Nutzbarkeiten abzuleiten. Im Gegensatz zu der vorliegenden Arbeit sind also für die Angabe der Anforderungen weitergehende Kenntnisse der eingesetzten Verfahren

und der resultierenden Nutzbarkeiten erforderlich. Andernfalls können die Nutzbarkeiten der generierten Pseudonyme nicht abgeleitet werden.

3.1.3 Anforderungsformulierung durch Angabe der Sicherheitsstufe

Eine weitere Methode der Formulierung von Anforderungen ist die Angabe der gewünschten Sicherheitsstufe. Bkakria et al. definieren in ihrer Arbeit [133] zur Pseudonymisierung von zu teilenden Datenbanken die vier Sicherheitsstufen randomisiert, deterministisch, ordnungserhaltend und homomorph. Auf diesen Sicherheitsstufen ist eine Ordnung formuliert. Durch die Auswahl von Pseudonymisierungsverfahren, die den geforderten Sicherheitsstufen entsprechen, werden die Anforderungen umgesetzt. Zu beachten ist, dass das Paillier-Verfahren sowohl randomisiert als auch homomorph ist. Jedoch wird das Verfahren für die Umsetzung der Anforderung homomorph verwendet. Aus den Sicherheitsstufen leiten die Autoren ab, welche Nutzbarkeiten in Form von Datenbankoperationen auf den pseudonymisierten Daten möglich sind. Eine Datenbankabfrage, die eine niedrigere Sicherheitsstufe fordert als durch die Pseudonymisierung umgesetzt, kann nicht beantwortet werden.
Ähnlich wie die vorliegende Arbeit, basiert die Arbeit von Schaad und Bkakria auf zum Zeitpunkt des Verfassen der Arbeit kryptographisch sicheren Primitiven. Im Gegensatz zu der vorliegenden Arbeit muss der Nutzer der Politik in der Lage sein, die zu ermöglichenden Operationen aus den Sicherheitsstufen abzuleiten. Auch werden in der vorliegenden Arbeit die Vertraulichkeitsanforderungen explizit formuliert. Die vorliegende Arbeit behandelt semistrukturierte textuelle Daten. Entsprechend beschränken sich die beschriebenen Nutzbarkeitsanforderungen nicht auf datenbankspezifische Operationen.

In diesem Abschnitt wurde der Stand der Wissenschaft zur Formulierung von Anforderungen an Pseudonymisierungsverfahren dargestellt. Hierfür wurden Beispielarbeiten identifiziert und beschrieben, in denen Anforderungen an die Nutzbarkeit und Vertraulichkeit indirekt oder direkt formuliert wurden. Es wurde festgestellt, dass dieselben Anforderungen vielfach unterschiedlich formuliert wurden. Zusammenfassend konnte im Vergleich festgestellt werden, dass durch die Anforderungsklassen der vorliegenden Dissertation eine Systematisierung und Zusammenführung von Anforderungsformulierungen dargestellt wird. Sie ermöglicht die Angabe von Anforderungen ohne Expertenwissen und erleichtert die Nachvollziehbarkeit der Nutzbarkeiten und die Auswahl von maßgeschneiderten Nutzbarkeiten durch Vorgabe von Anforderungsklassen.

3.2 Anforderungsklassen für Nutzbarkeitsanforderungen

Um im Rahmen der Anforderungen des Anwendungsfalls verarbeitbar zu sein, müssen Pseudonymisierungen Eigenschaften der zugrundeliegenden Klartextdaten erhalten. Zu diesem Zwecke werden Nutzbarkeitsanforderungen formuliert. In diesem Abschnitt werden daher zunächst Klassen von Nutzbarkeitsanforderungen an Pseudonymisierungen systematisiert aufgezeigt. Für die Ermittlung und Systematisierung der Anforderungen wurden verschiedene Datenanalyse-Tools

und Algorithmen gesichtet. Insgesamt wurden für die Nutzbarkeit von Pseudonymisierungen vier grundlegende Anforderungsklassen identifiziert. Diese sind die Klassen **Aufdeckbarkeit, Verkettbarkeit bzgl. einer Relation, Operation** und **Algorithmus**.
Ausgehend von den identifizierten Anforderungsklassen soll in Kapitel 6 eine gezielte Übersetzung von Anforderungen in diese als Nutzbarkeiten umsetzende Pseudonymisierungsverfahren genutzt werden. Zur Konstruierung von Pseudonymisierungen werden die so erzeugten Pseudonyme als Utility-Tags in einer Utility-Tags-Struktur aufgefasst. Hierbei ist ein Utility-Tag $u_{nu}(d)$ ein aus dem zugrundeliegenden Klartextdatum d aus der Datensammlung D mittels eines Pseudonymisierungsverfahrens erzeugtes Datum, das einer dedizierten Nutzbarkeitsanforderung nu genügt. Ein Pseudonym $p(d)$ eines Klartextdatums d ist dann die Menge aller diesem Klartextdatum zuordenbaren Utility-Tags innerhalb einer Pseudonymisierung $P(D)$ der Datensammlung D.
Eine Nutzbarkeitsanforderung wird auf einer Menge D von Klartextdaten definiert. Dies erlaubt eine kompakte Darstellung der Anforderung in der Beschreibungssprache `Util`[2] und eine stringente Ableitung der anzuwendenden Pseudonymisierungsverfahren[3] in Übersetzungsregeln[4]. Zu beachten ist die Folge, dass ein Pseudonym $p(d)$ aus einer Reihe von Utility-Tags $u_{nu}(d)$ bestehen kann, die zu verschiedenen Nutzbarkeitsanforderungen nu korrespondieren. In Kapitel 6 wird die Pseudonymisierung mit Utility-Tags beschrieben.
Im Folgenden werden die Anforderungsklassen für die Nutzbarkeiten beschrieben.

3.2.1 Anforderungsklasse Aufdeckbarkeit

Die Nutzbarkeitsanforderung der Aufdeckbarkeit ist absolut. Sie fordert die vollständige Freigabe des zugrundeliegenden Klartextdatums.

Definition 59: Aufdeckbarkeit
Ein Pseudonym $p(d)$ eines Klartextdatums d genügt der Nutzbarkeitsanforderung Aufdeckbarkeit, wenn in $p(d)$ mindestens ein Utility-Tag enthalten ist, das unter Nutzung von Zusatzinformation das Klartextdatum d freigibt.

Die für die Aufdeckbarkeit zu nutzende Zusatzinformation erlaubt den Zugriff auf das gesamte, dem Pseudonym zugrundeliegende Klartextdatum. Daher sind zum einen bei der Umsetzung Verfahren zu wählen, bei denen die Zusatzinformation leicht eingegrenzt werden kann. Dies ist bei Verschlüsselungsverfahren nach dem Stand der Wissenschaft der Fall. Die Zusatzinformation entspricht hierbei dem Entschlüsselungsschlüssel des Verfahrens. Der Zugriff auf die Zusatzinformation ist so zu schützen, dass Unbefugte diese nicht zur unintendierten Aufdeckung nutzen können.

[2] Siehe Kapitel 4.
[3] Siehe Kapitel 5.
[4] Siehe Kapitel 6.

3.2.2 Anforderungsklasse Verkettbarkeit bzgl. Relation

Um über Pseudonyme den Zusammenhang von zugrundeliegenden Klartextdaten in Bezug auf eine bestimmte Relation r zu ermitteln, wird die Nutzbarkeitsanforderung Verkettbarkeit bzgl. r definiert. Hierbei ist r eine mindestens zweistellige Relation, die über der Menge der Klartextdaten definiert ist. Für ihre Berechnung auf der Ebene der Pseudonyme müssen Utility-Tags existieren, die die erforderliche Aussage über die zugrundeliegenden Klartextdaten erlauben.

Definition 60: Verkettbarkeit bzgl. Relation r
Eine Menge P von Pseudonymen genügt der Nutzbarkeitsanforderung Verkettbarkeit bzgl. einer festgelegten Relation r, wenn jedes der in P enthaltenen Pseudonyme ein Utility-Tag enthält, das freilegt, ob zugrundeliegende Klartextdaten in Relation r zueinander stehen. Hierbei kann die Anwendung von Zusatzinformation erforderlich sein.

Der Berechnungsweg und das Ergebnis der Auswertung einer Verkettbarkeit bzgl. einer Relation r auf Pseudonymen muss nicht unbedingt identisch mit den Berechnungen zur Auswertung von r auf den zugrundeliegenden Klartextdaten sein. Daraus folgt, dass auf bezüglich einer Relation r verkettbaren Pseudonymen $\{p_i := p(d_i) \mid i \in \{1, \dots n\}\}$ eine Relation r' berechnet werden kann, für die gilt:

$$(p_i, p_j) \in r' \Longleftrightarrow (d_i, d_j) \in r \forall i, j \in \{1, \dots, n\} \tag{3.1}$$

Das Ergebnis der Auswertung auf den Pseudonymen wird dann so interpretiert, dass eine Aussage über die Relation r der definierten Verkettbarkeitsanforderung auf den zugrundeliegenden Klartextdaten getroffen werden kann.
Ein Beispiel für eine Ausprägung dieser Anforderungsklasse ist die Verkettbarkeit bezüglich der Relation Kleiner-Gleich. Hier wird für gegebene Klartextwerte a und b bestimmt, ob $(a, b) \in r$ (d.h. $a \leq b$). Für diese Nutzbarkeitsanforderungen wird ein auf ordnungsoffenbarenden Verschlüsselungsverfahren (engl. order-revealing encryption [25]) basierendes Pseudonymisierungsverfahren auf den Klartextdaten angewendet. Auf den auf diese Weise konstruierten Pseudonymen wird die Relation $\leq$ mit zur Konstruktion passenden Verfahren ausgewertet. Das Pseudonymisierungsverfahren und das zugehörige Auswertungsverfahren sind in Abschnitt 5.2.2 beschrieben.
Ein weiteres Beispiel ist die Verkettbarkeit bezüglich der Elementrelation. Hier wird für ein gegebenes Klartextdatum d und eine gegebene Menge M bestimmt, ob $d \in M$ gilt. Für diese Nutzbarkeitsanforderungen wird ein auf Bloom-Filtern basierendes Pseudonymisierungsverfahren [91] auf den Klartextdaten angewendet. Das Pseudonymisierungsverfahren und das zugehörige Auswertungsverfahren ist in Abschnitt 5.2.3 beschrieben.
Ein Spezialfall der Verkettbarkeit bzgl. einer Relation r ist die Pseudonym-Klartext-Verkettbarkeit. Hierbei wird ein Klartextdatum in eine Repräsentation überführt, in der es mit für die Verkettbarkeit bzgl. einer Relation r pseudonymisierter Daten verkettet werden kann. Hierdurch ergeben sich Unterschiede in der Umsetzung auf dem die Pseudonymisierung verarbeitenden System. Die Effekte werden in den Kapiteln 4, 5 und 6 aufgegriffen.

3.2.3 Operation

Wenn das Ergebnis einfacher Berechnungen auf Klartextdaten erforderlich ist, müssen die Pseudonyme Utility-Tags enthalten, die die Durchführung von Operationen erlauben, deren Ergebnis eine Offenlegung des Ergebnisses auf den Klartextdaten ermöglicht.

Definition 61: Operation
Eine Menge P von Pseudonymen genügt der Nutzbarkeitsanforderung Operation $*$, wenn jedes der in P enthaltenen Pseudonyme ein Utility-Tag enthält, das unter der Anwendung von Zusatzinformation das Ergebnis der Anwendung von $*$ auf die P zugrundeliegenden Klartextdaten freigibt.

Das Ergebnis der Anwendung von $*$ muss nicht notwendigerweise aus der Berechnung von $*$ erfolgen. Homomorphe Verfahren [1] ermöglichen zum Beispiel, auf dem Raum der Pseudonyme Berechnungen durchzuführen, deren Berechnungsergebnis äquivalent zur Anwendung von $*$ auf den zugrundeliegenden Klartextdaten sind.
Ein Beispiel für die Anforderungsklasse Operation ist die Nutzbarkeitsanforderung Addition. Um geeignete Pseudonyme zu generieren, wird ein auf dem additiv-homomorphen Paillier-Verschlüsselungsverfahren basierendes Pseudonymisierungsverfahren genutzt. Die Auswertung der Nutzbarkeit erfolgt über eine modulare Multiplikation auf einer ausreichend großen Gruppe. Pseudonymisierungs- und Auswertungsverfahren für diese Nutzbarkeit werden in Kapitel 5 als Algorithmen 11 und 12 beschrieben.
Für die Auswertung der Nutzbarkeit können zwei Varianten unterschieden werden. Zum einen kann die Verkettbarkeit als Pseudonym-Pseudonym-Verkettbarkeit ausgewertet werden. Hierbei wird untersucht, ob für gegebene Pseudonyme die zugrundeliegenden Klartexte in Relation r zueinander stehen. Die Pseudonyme sind Elemente einer Pseudonymisierung, die für die Nutzbarkeit der Verkettbarkeit bezüglich einer Relation r erstellt wurde. Zum anderen kann die Verkettbarkeit als Pseudonym-Klartext-Verkettbarkeit ausgewertet werden. Um zu bestimmen, ob das gegebene Klartextdatum mit dem einem Pseudonym zugrundeliegenden Klartextdatum in Relation steht, wird als Vorstufe der Berechnung das Klartextdatum in eine pseudonymisierte Repräsentation überführt. Hierfür ist typischerweise die Anwendung und entsprechend Kenntnisnahme der Zusatzinformation erforderlich, die auch bei der Erstellung der Pseudonymisierung als Parametrisierung des Pseudonymisierungsverfahrens zum Einsatz kommt. Dieser Umstand muss insbesondere dann berücksichtigt werden, wenn die Zusatzinformation zur Aufdeckung der Klartexte genutzt werden kann. Ist der zu verkettende Klartext in die pseudonyme Repräsentation überführt, kann die Auswertung der Verkettbarkeit analog zur Pseudonym-Pseudonym-Verkettbarkeit erfolgen.

3.2.4 Algorithmus

Wenn eine feste Abfolge verschiedener Operationen auf Pseudonymen ausgeführt werden soll, spricht man von der Anforderung Algorithmus.

Definition 62: Algorithmus
Eine Menge P von Pseudonymen genügt der Nutzbarkeitsanforderung Algorithmus für den Algorithmus $\mathcal{A}$, wenn unter Verwendung von Zusatzinformation die Freigabe des Ergebnisses der Berechnung von mindestens einer festgelegten Implementierung von $\mathcal{A}$ auf den Pseudonymen möglich ist.

Die Implementierung von $\mathcal{A}$ muss nicht notwendigerweise exakt mit der Abfolge der Operationen übereinstimmen, die durch $\mathcal{A}$ auf den den Pseudonymen in P zugrundeliegenden Klartextdaten ausgeführt worden wäre. Vielmehr handelt es sich bei der Implementierung von $\mathcal{A}$ um eine Abfolge von Operationen, deren Ergebnis eine ausreichend gute Näherung an die Ergebnisse ermöglicht, die $\mathcal{A}$ auf den Klartextdaten induziert hätte.
Gegen eine eigenständige Anforderungsklasse für diese Nutzbarkeitsanforderung könnte man argumentieren, dass die Anforderung Algorithmus eine mehrfache Ausführung der Anforderung Operation mit verschiedenen Operationen $*$ ist. Jedoch wird in Kapitel 5 dieser Arbeit dargelegt, dass die beiden Anforderungen auf der Ebene der Auswertung der Nutzbarkeit strukturell unterschiedlich sind. Um die in den Pseudonymisierungsverfahren und den damit einhergehenden Auswertungen begründeten Unterschiede zu kapseln und für den Nutzer transparent zu halten, wird die Definierung einer eigenständigen Anforderungsklasse als sinnhaft bewertet.
Ein Beispiel für diese Anforderungsklasse ist die Nutzbarkeitsanforderung der Auswertung des k-Means-Clusteringverfahrens auf pseudonymisierten Daten. Für k-Means existieren unterschiedliche PET-Varianten. Eine Variante, die ohne eine Verteilung der zu analysierenden Daten auf mehrere Parteien auskommt, wird in Kapitel 5 in Abschnitt 5.4.1 beschrieben.

3.3 Vertraulichkeitsanforderungen

Typischerweise bieten nutzbarkeitserhaltende Pseudonyme Einblicke in Teile der den zugrundeliegenden Klartextdaten inhärenten Informationen. Da diese Daten in der Regel weiterhin einer bestimmten, wenn auch nicht identifizierten Person zugeordnet werden können, sind sie als personenbezogene Daten zu betrachten. Gemäß den geltenden Datenschutzbestimmungen gilt es also, auch pseudonymisierte Daten mit der für personenbezogene Daten vorgesehenen Vorsicht zu verarbeiten [55].
Auch wenn Pseudonymisierungsverfahren eine Informationsreduktion bewirken und daher bzgl. der Reidentifizierbarkeit Betroffener risikomindernd sind, so kann ein Erfolg von Angriffen auf pseudonymisierte Daten zur Inferierung von Klartextdaten nicht ausgeschlossen werden.
Zum Schutz Betroffener vor Reidentifizierung bzw. Kenntnisnahme sensibler Information durch die Auswertung von Pseudonymen gehört die laut der Definition der Pseudonymisierung in der DSGVO geforderte Eigenschaft, dass Pseudonyme ohne Hinzunahme von sog. Zusatzinformation keine Rückschlüsse auf die zugrundeliegenden Klartextdaten zulassen dürfen. Daher bedarf es technisch-organisatorischer Maßnahmen, die den Zugriff und die Nutzung der Pseudonyme reglementieren. Diese ergeben sich aus der Umsetzung der Datenschutzprinzipien. Der Schutz der Vertraulichkeit der Klartextdaten bei der Verarbeitung der Pseudonyme soll so gestärkt werden. Daher werden sie bei der Formulierung der Vertraulichkeitsanforderungen an Pseudonyme berück-

sichtigt. Ausgehend von den Nutzbarkeitsanforderungen werden Pseudonymisierungen erstellt. Diese sind als Sammlungen von Pseudonymen zu sehen. Kombiniert mit auf den Pseudonymisierungen umgesetzten Vertraulichkeitsanforderungen soll das Risiko der unerlaubten Reidentifizierung weiter gemindert und der Schutz der Betroffenen sichergestellt werden.
Im Folgenden werden implizite und explizite Vertraulichkeitsanforderungen unterschieden.

3.3.1 Implizite Vertraulichkeitsanforderungen

Implizite Anforderungen fließen in den Entwurf von Privacy-Enhancing-Technologies ein und sind maßnahmenübergreifend umzusetzen. Sie erfordern in der Regel keine weitere Parametrisierung.

Definition 63: Beschränkung der Reidentifizierbarkeit Betroffener
Die Reidentifizierung von Subjekten aus einem Pseudonym soll nur dann ermöglicht sein, wenn diese zur Erfüllung des Datenverarbeitungszwecks unbedingt erforderlich ist.

Aus dieser Anforderung folgt, dass insbesondere die explizite oder implizite Aufdeckung von Klartextdaten aus Pseudonymen immer dann vermieden werden muss, wenn sie nicht unbedingt erforderlich ist. Dies gilt insbesondere dann, wenn die den Pseudonymen zugrundeliegenden Klartextdaten als sensibel eingestuft wurden.

Definition 64: Datenminimierung
Pseudonyme sollen so wenig personenbezogene Information wie möglich preisgeben. Die preisgegebene Information soll möglichst dem für die Erfüllung des Datenverarbeitungszweck erforderlichen Umfang entsprechen. Darüber hinaus soll möglichst keine personenbezogene Information abfließen.

Die Datenpseudonymisierung ist bereits eine Technik zur Informationsreduktion. Um die Vertraulichkeitsanforderung der Datenminimierung zu erfüllen, ist jedoch zu beachten, dass auch pseudonymisierte Daten nur im unbedingt erforderlichen Maße verarbeitet werden. Personenbezogene Daten, die in der Verarbeitung nicht unbedingt benötigt werden, sollen auch nicht in pseudonymisierter Form verarbeitet werden. Weiterhin gehört zur Datenminimierung, dass Pseudonyme möglichst nur die Nutzbarkeitsanforderungen erfüllen, die für die Erfüllung des Datenverarbeitungszwecks unbedingt erforderlich sind. Bei der Umsetzung ist auch gesetzlich verankert, dass der Stand der Wissenschaft und Technik zu beachten ist.[55]

Definition 65: Mitigation und Risikominimierung
Wann immer möglich, sind zusätzliche Maßnahmen zu ergreifen, die zu einer weiteren Minderung des Restrisikos der Reidentifizierbarkeit bzw. Preisgabe personenbezogener Information aus Pseudonymen führt.

Eine Risikominimierung ist hierbei im Rahmen der nach Stand der Wissenschaft und Technik verfügbaren technisch-organisatorischen Maßnahmen zu verstehen. Techniken zur exakten Eingrenzung von Eigenschaften erhaltenden Nutzbarkeitsanforderungen, die ausschließlich auf bestimmte

Information hinweisen und das Induzieren weiterer Information auch unter Nutzung von Zusatzinformation beweisbar ausschließen, sind bisher nicht bekannt. Daher kann es ohne das Datum zu löschen bisher keine absolute bzw. beweisbare Risikominimierung geben.

Definition 66: Datentrennung
Pseudonyme sollen getrennt von Zusatzinformation gespeichert werden, wenn diese Zusatzinformation für die Aufdeckung des zugrundeliegenden Klartextdatums erforderlich ist.

Das Konzept der Datentrennung soll die Aufhebung einer erfolgten Datenpseudonymisierung bzw. das Inferieren von Information aus den Pseudonymen weiter eingrenzen und für Unbefugte erschweren. Zusatzinformation kann z.B. in besonders geschützten Umgebungen verarbeitet und gespeichert werden. Mögliche Angriffe auf das System haben dann mit größerer Wahrscheinlichkeit in geringerem Ausmaße einen Abfluss personenbezogener Information zur Folge.

3.3.2 Explizite Vertraulichkeitsanforderungen

Explizite Anforderungen erfordern die Angabe weiterer Information zum Zeitpunkt der Umsetzung.

Definition 67: Umgang mit sensiblen personenbezogenen Daten
Pseudonyme, die aus sensiblen Klartextdaten erzeugt wurden, sollten als solche erkennbar sein und mit entsprechender Vorsicht verarbeitet werden.

Als sensibel eingestufte personenbezogene Daten erleichtern die Reidentifizierung Betroffener und geben mitunter intime persönliche Information preis. Beispiele sind die Ethnizität und bestimmte diagnostizierte Krankheiten. Daher müssen sie nach Artikel 9 der DSGVO auch dann unter Beachtung besonderer Schutzmaßnahmen verarbeitet werden, wenn sie in pseudonymisierter Form vorliegen. Solche Pseudonyme sind daher im verarbeitenden System entsprechend zu kennzeichnen. Entsprechend gekennzeichnete Pseudonyme können zum Beispiel von einer bestimmten Datenverarbeitung ausgeschlossen werden.

Definition 68: Begrenzung der Speicherungsdauer
Pseudonyme sollen nur solange gespeichert werden, wie deren Nutzung zur Erfüllung des Verarbeitungszwecks dies unbedingt erfordert.

Die Begrenzung der Speicherungsdauer kann durch technisch-organisatorische Maßnahmen umgesetzt werden. Dies kann ausgehend von einer definierten Speicherdauer eines Pseudonyms geschehen.

Definition 69: Rollenbindung
Pseudonyme sollten so geschützt werden, dass die Nutzung auf Subjekte beschränkt ist, denen im System eine festgelegte Rolle zugewiesen ist.

Die Rollenbindung ermöglicht, Vertrauen auf Basis von Rollen in einem System zu definieren. Zum Beispiel kann in einem System festgelegt werden, dass die Rolle *Richter* vertrauenswürdiger als die Rolle *Kodierkraft* ist und damit grundsätzlich Zugriff auf bestimmte Dokumente erhalten soll. Hintergrund ist, dass bestimmte Rollen im System eine begründete Erlaubnis zur Verarbeitung personenbezogener Daten auch unabhängig vom dokumentierten Zweck haben können. Als Alternative zur Zweckbindung und Ergänzung dieser ermöglicht die Rollenbindung daher die Verarbeitung eines Pseudonyms für bestimmte Subjekte.

Definition 70: Zweckbindung
Pseudonyme sollten so geschützt werden, dass die Nutzung auf bestimmte, nachweisbare Zwecke beschränkt ist.

Die Zweckbindung bildet die in der DSGVO geforderte, gleichnamige Anforderung ab. Hierbei soll durchgesetzt werden, dass personenbezogene Daten ausschließlich für Zwecke verarbeitet werden, die gemäß der DSGVO zum Erhebungszweck kompatibel sind.

3.4 Designentscheidungen

In diesem Kapitel wird ein Anforderungsmodell für Pseudonymisierungen erarbeitet. Beim Entwurf der Anforderungsklassen werden mehrere Designentscheidungen getroffen. Diese werden im Folgenden begründet.

3.4.1 Datenadressierung

Bei der Adressierung der Daten zur Beschreibung der Anforderung an deren Pseudonyme wurde eine nutzbarkeitsorientierte Vorgehensweise gewählt. Daraus folgt, dass für jede Nutzbarkeitsanforderung alle von dieser Anforderung betroffenen Daten angegeben werden. Der Vorteil ist hierbei, dass durch die Bündelung von Klartextdaten in einer Nutzbarkeit die betroffenen Daten leicht identifiziert werden können. So kann nachvollzogen werden, für welche Klartextdaten mehrfach Nutzbarkeitsanforderungen formuliert wurden. Auch kann nachvollzogen werden, wenn eine Nutzbarkeitsanforderung über mehrere Attribute formuliert wird. Soll die Pseudonymisierung auf das Risiko der Reidentifizierbarkeit untersucht werden, so sind eventuelle Wechselwirkungen der Nutzbarkeiten und deren Einfluss auf die mögliche Nutzung zur Reidentifizierung durch den Einfluss bestimmter Daten leicht nachvollziehbar.

3.4.2 Explizite Vertraulichkeitsanforderung als Anhang von Nutzbarkeitsanforderungen

Bei der Formulierung von Anforderungen dieser Klasse wurde entschieden, diese stets als Einschränkung einer bestimmten Nutzbarkeitsanforderung zu formulieren. Entsprechend werden explizite Vertraulichkeitsanforderungen nicht als eigenständige Anforderungen an bestimmte

Klartextdaten formuliert. Vielmehr werden sie als Einschränkung einer Nutzbarkeitsanforderung festgelegt, die für bestimmte, zu pseudonymisierende Klartextdaten formuliert wurde. Bei der Umsetzung der Datenpseudonymisierung gelten sie daher für die Pseudonyme aller Daten, die für die Nutzbarkeitsanforderung formuliert wurden, zu der als Einschränkung eine entsprechende Vertraulichkeitsanforderung formuliert wurde. Diese Anforderungen umfassen den Umgang mit sensiblen personenbezogenen Daten, die Begrenzung der Speicherungsdauer sowie die Rollenbindung und die Zweckbindung.

3.4.3 Implizite Vertraulichkeitsanforderungen

Ein Design-Ziel ist es, möglichst wenige Vertraulichkeitsanforderungen explizit zu formulieren. Damit sollen Anwender ohne Expertenkenntnisse die Anforderungen möglichst intuitiv und selbsterklärend formulieren können. Daher wurden Vertraulichkeitsanforderungen nach Möglichkeit als implizit betrachtet. Dies wurde für die Anforderungen Beschränkung der Reidentifizierbarkeit Betroffener, Datenminimierung, Mitigation und Risikominimierung und Datentrennung als möglich erachtet. Diese Anforderungen fließen in das Design der Pseudonymisierungsverfahren und der Übersetzungsregeln ein. Auf dem die Pseudonymisierungen verarbeitenden System fließen sie durch das Ergreifen weiterer technisch-organisatorischer Maßnahmen ein. Jedoch erfordern sie keine Angabe zusätzlicher Information wie etwa ein Datum zur Ermittlung der Speicherungsdauer.

3.4.4 Aufdeckbarkeit als eigene Anforderungsklasse

Die Aufdeckung eines Klartextdatums $d \in D$ aus einem Pseudonym $p(d)$ kann als Relation R_{discl} aufgefasst werden:

$$R_{discl} = \Big\{(p(d), d) \mid p(d) \in P(D), d = p^{-1}(p(d))\Big\}.$$

Würde die Aufdeckbarkeit als Relation aufgefasst und im Rahmen der entsprechenden Anforderungsklasse definiert werden, so müsste der Anwender die Umkehrungsvorschrift $p^{-1} : P \longrightarrow D$ des zugrundeliegenden Pseudonymisierungsverfahrens $p : D \longrightarrow P$ kennen und als entsprechende Verkettbarkeitsrelation r angeben. Damit wäre die Ebene der Definierung der Anforderung konzeptionell nicht mehr von der Ebene der Auswahl und der Parametrisierung der Pseudonymisierungsverfahren[5] getrennt. Um das Pseudonymisierungsverfahren von der Ebene der Anforderungsdefinierung zu trennen, wurde daher die eigenständige Anforderungsklasse Aufdeckbarkeit definiert.

Die Auswertung dieser Nutzbarkeitsanforderung hat eine vollständige Verfügbarmachung des Klartextdatums zur Folge. Dies entspricht dem höchsten Maß an Kritikalität, den ein nutzbarkeitserhaltendes Pseudonym bezüglich des Schutzes der Vertraulichkeit der zugrundeliegenden Klartextdaten aufweisen kann. Mit der Definierung der Aufdeckbarkeit als eigenständige Anforderungsklasse wird ein erhöhtes Bewusstsein bei dem Anwender bezweckt.

[5] siehe Abbildung 5 zum Rahmenwerk mit Unterteilung der Komponenten nach Betrachtungsebenen.

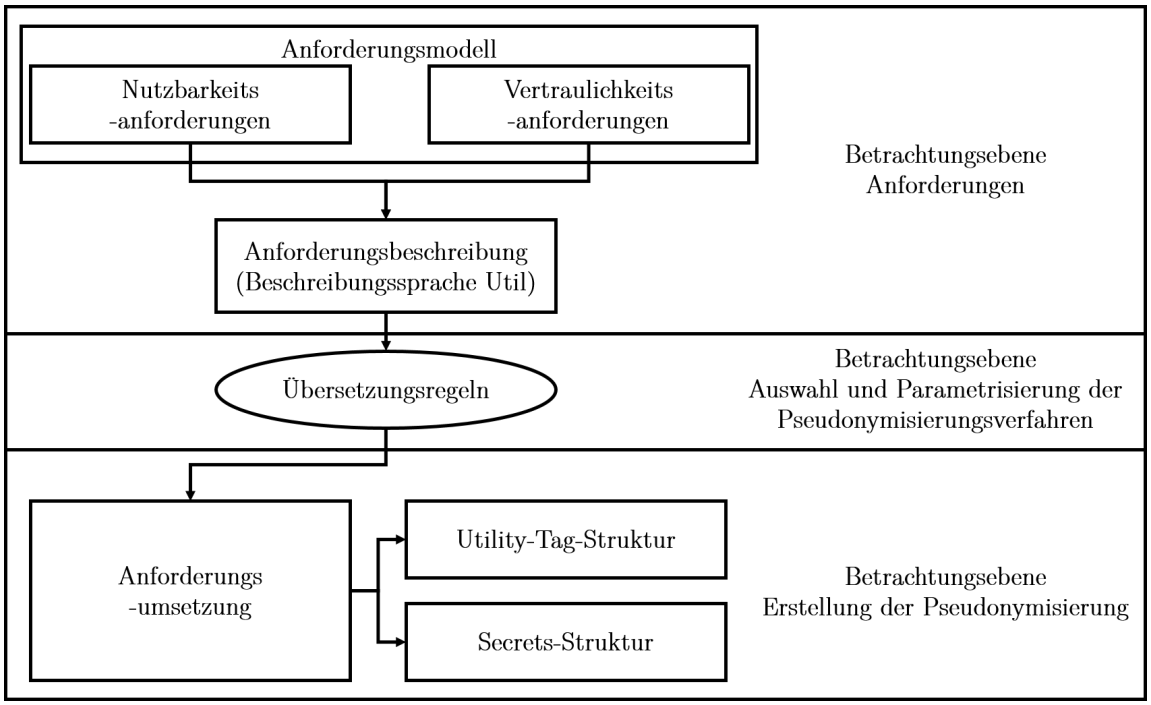

Abbildung 5: *Die Hauptkomponenten des Rahmenwerks mit einer Unterteilung der Komponenten in Betrachtungsebenen.*

3.4.5 Operationen als eigene Anforderungsklasse

Die Ausführung von einfachen wiederkehrenden Operationen wie die Addition, die Multiplikation und die Durchschnittsbildung kommt in der Anwendung entweder als Teil eines Algorithmus oder alleinstehend vor. Alleinstehend kann eine Operation einen Analyseschritt in einer Analyse-Pipeline, d.h. einer zusammengesetzten Abfolge von auf den Daten auszuführenden Routinen darstellen. Dieser Analyseschritt kann wiederkehrend ausgeführt werden. Eine Operation $o : D \longrightarrow W$ operiert auf einem Definitionsbereich D und bildet dessen Werte in den Wertebereich W ab. Daher kann eine Operation als l-stellige Relation R_o dargestellt werden:

$$R_o = \{\{(d_1, \ldots, d_{l-1}, w) \mid d_i \in D, l \geq 2, w = o(d_1, \ldots, d_l) \in W\}.$$

Eine zunächst naheliegende Designentscheidung wäre, Operationen als Nutzbarkeitsanforderung stets als Relationen aufzufassen. Dementsprechend würden Operationen als Variante der Anforderungsklasse Verkettbarkeit betrachtet werden. In dieser Anforderungsklasse sind die Relationen mit beliebiger, aber fester Stelligkeit zu definieren. Für als Relationen darzustellende Operationen hätte dies zur Folge, dass eine mehrfache Ausführung einer Operation umständlich als Verknüpfung identischer Relationen modelliert werden müsste. Ist die Ausführung der Operation z.B. beliebig oft erlaubt, so müsste die Relation umständlich mittels Abbildungen unterschiedlicher Stelligkeit oder rekursiv nachgebildet werden. Dies hätte einen zusätzlichen Aufwand für den Anwender zur Folge. Zusätzlich würde dem Nutzer die intuitive Nutzung mathematischer Operationen verwehrt

bleiben. Daher wurde entschieden, für einfache Operationen eine eigene Anforderungsklasse zu definieren. Hierzu zählen die Addition und die Multiplikation.

3.4.6 Algorithmen als eigene Anforderungsklasse

Die Ausführung von Algorithmen auf pseudonymisierten Daten entspricht einer Abfolge einer endlichen Menge von Operationen. Daher ist es zunächst naheliegend, Algorithmen als eine Menge von Nutzbarkeitsanforderungen der Anforderungsklasse Operation aufzufassen. Dies hätte die Herausforderung für den Anwender zur Folge, bei der Definierung der Nutzbarkeitsanforderungen alle auszuführenden Algorithmen als Abfolge der enthaltenen Operationen zu definieren. Gerade bei komplexen, vielschrittigen Algorithmen wäre dies mit einem erhöhten Aufwand verbunden.
Ein weiterer entscheidender Nachteil dieser Vorgehensweise wäre, dass eine Einschränkung der Nutzbarkeiten auf genau den Umfang der im Algorithmus erforderlichen Operationen kaum möglich wäre. Ist zum Beispiel die Anzahl der Iterationen und damit die Anzahl der Ausführung einer bestimmten Operation eines Algorithmus abhängig von der Erreichung eines zum Berechnungszeitpunkt zu bestimmenden, datenabhängigen Schwellwerts, so kann diese Anforderung nur sehr umständlich und wenig intuitiv abgebildet werden. Daher könnte hier kaum eine Einschränkung der Auswertung der Nutzbarkeit vorgenommen werden. Dies hat zur Folge, dass möglicherweise deutlich mehr Nutzbarkeiten auf den generierten pseudonymisierten Daten zur Verfügung stehen als für die Auswertung des Algorithmus unbedingt erforderlich. Eine Ausnutzung dieser zusätzlichen Nutzbarkeiten hätte möglicherweise die Aufdeckung der pseudonymisierten Daten zur Folge. Ein Beispiel ist das k-Means-Clustering-Verfahren[6]. Um es als Abfolge von Operationen abzubilden, müssten die für die Distanzberechnung und die Berechnung der Clusterzentren die Nutzbarkeitsanforderungen der Operation Addition (und entsprechend die Subtraktion) und der Multiplikation (und entsprechend die Division) sowie für die Bestimmung des Abbruchkriteriums die Verkettbarkeit bezüglich der Relation Kleiner-Gleich auf dem gesamten Datensatz formuliert werden. Da die Entwicklung der Verteilung der Elemente über die Cluster vor Beginn der Iterationen der Clusterberechnungen nicht bekannt sind, müssten die genannten Nutzbarkeitsanforderungen auf der gesamten Datensammlung formuliert werden. Dies hätte zur Folge, dass alle genannten Nutzbarkeiten beliebig oft ausgeführt werden können.
Um den Umfang der zur Verfügung stehenden Nutzbarkeiten von für die Erfüllung der Nutzbarkeitsanforderung eines Algorithmus pseudonymisierten Daten zu reduzieren, werden in der PET-Forschung Privatsphäre schützende Varianten von Algorithmen entwickelt. Beispiele sind Privatsphäre schützende Varianten maschineller Lernverfahren [29]. Um das Einbinden von Privatsphäre schützenden Varianten von Algorithmen in dem in der vorliegenden Arbeit vorgestellten Rahmenwerk zu ermöglichen, wurde eine eigenständige Anforderungsklasse Algorithmus eingeführt. In Kapitel 15 wird eine Privatsphäre schützende Variante des k-Means-Clustering-Verfahrens beschrieben.

[6] Siehe Kapitel 5.4.1

4 Beschreibungssprache für Nutzbarkeitspolitiken von Pseudonymisierungen

In der Dissertation wird aufgezeigt, wie die in Abschnitt 2.1 und in Kapitel 3 angeführten Anforderungen an Pseudonyme für eine automatisierte Aufbereitung von flexibel konfigurierbaren nutzbarkeitserhaltenden Pseudonymisierungen herangezogen werden können. Ziel der Erarbeitung der in Kapitel 3 vorgestellten Anforderungen an Pseudonyme ist die Erleichterung der automatisierten Erstellung von Pseudonymisierungen. Hierfür müssen die Anforderungen maschinenlesbar formuliert werden können. Dies wird mit der in diesem Kapitel vorgestellten Beschreibungssprache `Util` ermöglicht. In `Util` werden Anforderungen an eine Pseudonymisierung einer Datensammlung D als Nutzbarkeitspolitik formuliert. Die eingesetzten Pseudonymisierungsverfahren und deren Parametrisierung bleiben für die Formulierenden der Anforderungen transparent. Insgesamt soll mit `Util` ein Beitrag zur Erleichterung der Formulierung von Anforderungen an Pseudonymisierungen geleistet werden. Zusätzlich sollen in `Util` formulierte Politiken als Dokumentation der durch eine nutzbarkeitserhaltende Pseudonymisierung zur Verfügung gestellten Information dienen.
In diesem Kapitel wird zunächst in Abschnitt 4.1 der Stand der Wissenschaft der Politik-Sprachen mit Bezug zu Privacy erarbeitet. Dann werden in Abschnitt 4.2 mögliche Anwendungszwecke für die Formulierung und Speicherung einer `Util`-Nutzbarkeitspolitik beschrieben. In Abschnitt 4.3 wird der Entwurf und Aufbau der Beschreibungssprache `Util` hergeleitet. Schließlich werden in 4.4 Entscheidungen, die beim Entwurf von `Util` getroffen wurden, motiviert und erläutert.

Es werden die folgenden **fünf Anforderungen an eine Politik-Sprache** zur Formulierung von Politiken für die Nutzbarkeits- und Vertraulichkeitsanforderungen (Nutzbarkeitspolitik) an Pseudonymisierungen gestellt:

1. **Formulierbarkeit der Anforderungen:** Für eine sinnvolle Umsetzung der Nutzbarkeits- und Vertraulichkeitsanforderungen müssen diese in einer Politik formuliert werden können.
2. **Maschinenlesbarkeit:** Für eine automatisierte Ableitung von geeigneten Verfahren zur Umsetzung der Anforderungen in Pseudonymisierungen muss die Politik maschinenlesbar und entsprechend automatisiert verarbeitbar sein.
3. **Menschenlesbarkeit:** Um eine Nachvollziehbarkeit der in einer Pseudonymisierung umgesetzten Anforderungen zu gewährleisten, muss die Politik menschenlesbar sein. Als menschenlesbar wird im Folgenden eine Sprache bezeichnet, deren Elemente aus natürlicher

Sprache abgeleitet und für Menschen nachvollziehbar und verständlich formuliert sind und deren Strukturen für den Nutzer nachvollziehbar sind.

4. **Dokumentierbarkeit:** Damit eine Politik als Dokumentation der zwecks Erfüllung von Datenschutzanforderungen ergriffenen Pseudonymisierungsmaßnahmen genutzt werden kann, muss diese Elemente für die Umsetzung der Datenschutzprinzipien enthalten.
5. **Privacy-by-Design:** Eine Politik-Sprache sollte *by-Design* Privacy-respektierend sein. Dies beinhaltet die Berücksichtigung des Schutzes der Vertraulichkeit personenbezogener Daten bereits zum Zeitpunkt der Festlegung der Elemente und Strukturen der Politik-Sprache.

4.1 Stand der Wissenschaft

Politik-Sprachen zur Erstellung und Dokumentation von Vertraulichkeitsanforderungen sind in der Privacy-Forschung wohlbekannt. Die im Kontext der vorliegenden Arbeit relevanten Sprachen werden im Folgenden beschrieben. Dabei werden relevante Erkenntnisse aus der zuvor von der Autorin veröffentlichten Arbeit [88] rekapituliert und um Erkenntnisse aus den Übersichtsarbeiten von Kumaraguru et al. [97], Leicht et al. [99] und Morel et al. [112] ergänzt. Hierbei werden zunächst Ansätze gelistet, die den Anforderungen an eine Politik-Sprache für Nutzbarkeitspolitiken zumindest zum Teil genügen. Die Erfüllung dieser Anforderungen wird in Tabelle 1 zusammengefasst. Gleichzeitig wird untersucht, ob die in der Literatur beschriebenen Politik-Sprachen für die Formulierung von Nutzbarkeitspolitiken geeignet sind.

XACML[1] [141] ist eine Sprache für Zugriffskontrollpolitiken. Sie verfügt über eine Erweiterung für die Deklarierung des Zwecks eines Zugriffs. Die PrimeLife-Politik-Sprache PPL [145] baut auf XACML auf. Sie wird für die Festlegung von Zugriffskontrollregeln unter Nutzung von aus der Sicht des Systems anonymen Zugangsdaten verwendet. Die Erweiterung Accountability-PPL (A-PPL) [9] erlaubt eine Durchsetzung bestimmter weiterer Vertraulichkeitsanforderungen und eine weitreichende Nachvollziehbarkeit der auf personenbezogenen Daten ausgeführten Aktionen. Dies beinhaltet die Festlegung von in der Cloud relevanten Parametern. Beispiele sind die Einschränkung der Verfügbarkeit der Daten auf bestimmte Regionen und die Einschränkung des Zeitraums der Erlaubnis der Datennutzung für einen festgelegten Zweck. Diese miteinander verwandten Politik-Sprachen verfügen über die Möglichkeit der Angabe bestimmter Vertraulichkeitsanforderungen wie Zwecke und Begrenzung des Zeitraums, in dem ein Zugriff auf ein Datum ermöglicht wird. Jedoch sind die Entscheidungen des Zugriffs ausschließlich eine Freigabe von Daten. Um diese Sprachen für die Formulierung von Nutzbarkeits- mit Vertraulichkeitsanforderungen nutzen zu können, müssten sie um grundlegende Strukturen zur Beschreibung von Nutzbarkeitsanforderungen erweitert werden. Zusätzlich müssten weitere Vertraulichkeitsanforderungen formuliert werden. Die in den Sprachen bereits existierenden Strukturen für die Definierung der Zugriffskontrollregeln würden ungenutzt bleiben. Zusammenfassend wird festgestellt, dass XACML, PPL und A-PPL auch im Falle einer entsprechenden Erweiterung

[1] eXtensible Access Control Markup Language

Tabelle 1: *Zusammenfassung der Erfüllung der Anforderungen durch die einzelnen Politik-Sprachen. Legende: + bedeutet, dass die Anforderung erfüllt ist. ~ bedeutet, dass die Erfüllung der Anforderung eingeschränkt möglich ist. – bedeutet, dass die Erfüllung der Anforderung nicht möglich ist.*

	Anforderungen				
Politik-Sprache	(1) Formulierbarkeit	(2) Maschinenlesbarkeit	(3) Menschenlesbarkeit	(4) Dokumentierbarkeit	(5) Privacy-by-Design
A-PPL [9]	~	ja	+	~	+
AIR [92]	~	+	–	~	+
APPEL (P3P) [44]	~	+	+	~	+
EPAL [8]	~	+	+	+	+
E-P3P [8]	~	+	–	~	+
Jeeves [155]	~	+	+	~	+
LPL [68]	~	+	+	+	+
P2U [79]	~	+	+	~	+
P3P [44]	~	+	–	~	+
PPL [145]	~	+	+	~	+
SecPAL4P [14]	~	+	+	~	+
XPref [4]	~	+	+	~	+
Util [89]	+	+	+	+	+

eher für eine Implementierung von Zugriffskontrolle auf pseudonymisierten Daten geeignet sind. Weniger jedoch eignen sie sich für die Anforderungsformulierung an zu pseudonymisierende Daten und Dokumentation dieser Anforderungen. Weiterhin ist die Menschenlesbarkeit durch die existierenden Strukturen erschwert.
SecPAL [12][13] ist eine Sprache zur dezentralisierten Formulierung von Autorisierungsregeln. Mit ihr werden Subjekten Rechte in Form von Prädikaten für bestimmte Objekte zugewiesen. Die Erweiterung SecPAL4P [14] erlaubt die Formulierung weiterer Obligationen, die bei der Datennutzung berücksichtigt werden müssen. Ein Beispiel ist die Obligation, ein Datum nach einer festgelegten Speicherdauer zu löschen. Diese Obligationen setzen Vertraulichkeitsanforderungen in Teilen um. Ähnlich wie die zuvor beschriebenen Sprachen fehlen jedoch Strukturen zur Formulierung von Nutzbarkeitsanforderungen an zu pseudonymisierende Daten. In Abhängigkeit von der Auswertung der Autorisierungsregeln in Kombination mit den Obligationen werden Daten entweder im Klartext freigegeben oder eine Freigabe wird dem Subjekt verweigert. Da für die Daten über die Freigabe im Klartext hinaus keine Abstufungsmöglichkeit existiert, können Nutzbarkeitsanforderungen nicht abgebildet werden.

Weitere Politik-Sprachen zur Beschreibung von nachvollziehbarem Vertraulichkeitsschutz personenbezogener Daten durch Zugriffskontrolle wurden für bestimmte Anwendungen beschrieben. Für die Definierung von Privacy-Anforderungen an personenbezogene Daten im

Semantic Web wurde AIR [92] eingeführt. Für die Angabe von Nutzerpräferenzen bezüglich der Datenweitergabe im World Wide Web wurde P3P [44] formuliert. Es konnte mit existierenden Erweiterungen wie APPEL [44] und XPref [4] zur Einschränkung und Steuerung der Weitergabe bestimmter Nutzerinformation verwendet werden. Die Politik-Sprache P2U von Iyilade et al. [79] ermöglicht in diesem Kontext zusätzlich die Kontrolle der Datenweitergabe zur Erfüllung von Sekundärzwecken. EPAL [8] erlaubt die Festlegung von Regeln zur Nutzung von in Unternehmen gesammelten Daten. Die Formulierung von Vertraulichkeitsanforderungen ist eingeschränkt möglich. Analog zu einer Vielzahl der hier vorgestellten Sprachen kann lediglich festgelegt werden, ob ein Datum unter bestimmten Voraussetzungen im Klartext verfügbar gemacht werden darf oder nicht. Damit ist auch EPAL nicht für die Nutzung im Kontext der vorliegenden Arbeit geeignet.

Eine weitere Sprache, mit der die Vertraulichkeit personenbezogener Daten kontrolliert werden kann, ist Jeeves [155]. Als domänenspezifisch in Python implementierte Sprache ermöglichen mit Jeeves formulierte Politiken die Steuerung der Sichtbarkeit, der Übergabe und den Zugriff auf in der Politik definierte Variablen. Darüber hinaus verfügt Jeeves jedoch nicht über Strukturen, mit denen selektiv Nutzbarkeiten verfügbar gemacht werden können. Auch ist ein genaues Verständnis der Implikationen für die Reidentifizierbarkeit Betroffener des durch Jeeves-Politiken gesteuerten Informationsfluss innerhalb von Programmen erforderlich. Im Unterschied zu AIR, P3P und die Erweiterungen von P3P kann Jeeves in unterschiedlichen Python-Anwendungen umgesetzt werden.

Armin Gerl entwickelte mit LPL [68] eine Politik-Sprache, mit der gezielt mehrere Mechanismen zur De-Identifikation adressiert werden können. Diese erlauben eine zumindest implizite Umsetzung einzelner weniger Nutzbarkeiten wie der Aufdeckbarkeit und der Verkettbarkeit bezüglich der Relation Gleichheit. LPL verfügt by-Design über Strukturen zur Umsetzung von Vertraulichkeitsanforderungen nach den Datenschutzprinzipien. Ihre Erweiterungen [66, 67] erlauben die Definierung von aufdeckbaren Pseudonymen. Jedoch kann mit LPL keine gezielte Formulierung von Nutzbarkeitsanforderungen vorgenommen werden. Auch sind Kenntnisse der Pseudonymisierungsverfahren für die Beschreibung der Anforderungen in [67] erforderlich.
Zusammenfassend kann festgestellt werden, dass der Großteil der vorgestellten Politik-Sprachen Anforderung 1 (Formulierbarkeit der Anforderungen) rudimentär durch die Ermöglichung von Klartextfreigabe und der Formulierbarkeit von Vertraulichkeitsanforderungen erfüllen. Anforderung 2 (Maschinenlesbarkeit) wird von allen vorgestellten Sprachen erfüllt. Anforderung 3 (Menschenlesbarkeit) wird von allen Sprachen außer AIR, P3P und E-P3P erfüllt. Anforderung 4 (Dokumentierbarkeit) wird von EPAL und LPL erfüllt. Anforderung 5 (Privacy-by-Design) wird von allen vorgestellten Sprachen erfüllt.

Fazit Fazit für die vorliegende Arbeit ist, dass keine Sprache existiert, die ohne grundlegende Modifizierung für die Formulierung von Nutzbarkeitspolitiken genutzt werden kann. Hierzu müsste bei allen Sprachen die grundlegende Annahme geändert werden, dass die Steuerung der Verarbeitung eines Datums binär, d.h. entweder vollständig oder nicht ermöglicht wird. Bei LPL ist die

Ersetzung identifizierender Klartextdaten durch Pseudonyme möglich [66]. Auch ermöglicht LPL über eine Erweiterung die Möglichkeit, Anonymisierungstechniken auf den Daten auszuführen [66]. Jedoch beinhaltet auch LPL keine Möglichkeit zur Formulierung dedizierter Nutzbarkeitsanforderungen. Vielmehr können Nutzbarkeitsanforderungen implizit über die Angabe des einzusetzenden Pseudonymisierungsverfahrens formuliert werden. Selbst im Falle der Ermöglichung von Datenpseudonymisierung in einigen der untersuchten Politik-Sprachen ist der Zugriff auf ein Datum stets binär: Entweder wird die Verarbeitung eines Pseudonyms vollständig erlaubt oder gänzlich verwehrt. Daher wird im Folgenden die *Utility-Policy*-Sprache `Util`, eine Politik-Sprache für die Formulierung von Nutzbarkeitsanforderungen mit Vertraulichkeitsanforderungen in Form einer Politik für die nutzbarkeitserhaltende Pseudonymisierung beschrieben. `Util` wurde bereits in der Arbeit [89] vorgestellt.

4.2 Zweck einer Nutzbarkeitspolitik

Zusätzlich zu der Verwendung der in `Util` deklarierten Anforderungen an eine Politik-Sprache zur automatisierten Ableitung von Pseudonymisierungen soll eine Nutzbarkeitspolitik auch die aus den zugrundeliegenden Klartextdaten in einer Pseudonymisierung zur Verfügung gestellten Information dokumentieren. Weiterhin dokumentiert eine Nutzbarkeitspolitik die für den Schutz der Vertraulichkeit der referenzierten Daten eingesetzten technisch-organisatorischen Maßnahmen. Die Beschreibungssprache `Util` für Nutzbarkeitspolitiken ist eine Mensch-Maschine-Schnittstelle zur Beschreibung der Anforderungen, die eine geplante Datenpseudonymisierung erfüllen soll. Um diese Schnittstelle auch für Nichtexperten im Bereich der Privacy-Enhancing-Technologies (PET) zugänglich zu machen, wurde `Util` so entworfen, dass die Beschreibung der Nutzbarkeits- und der Vertraulichkeitsanforderungen mit geringen Kenntnissen der PET möglich ist. Insbesondere werden Kenntnisse in der Angewandten Kryptographie, wie sie für die Erstellung einer risikomindernden, gleichzeitig aber nutzbarkeitserhaltenden Pseudonymisierung erforderlich sind, nicht für die Angabe der Anforderungen in `Util` benötigt. Vielmehr wird angenommen, dass die Formulierung der für die zweckmäßige Verarbeitung der Daten erforderlichen Nutzbarkeiten ohne Kenntnisse der PET, jedoch unter Kenntnis der auf den pseudonymisierten Daten auszuführenden Operationen ohne eine steile Lernkurve möglich ist. Verglichen mit dem Erlernen der PET ist dies eine geringe Hürde für die Anwendung von nutzbarkeitserhaltenden Pseudonymisierungen zu sein. Nichtexperten wird somit die Erstellung von nutzbarkeitserhaltenden, maßgeschneiderten Pseudonymisierungen erleichtert. Somit wird insgesamt ein Beitrag zur Verbreitung der Nutzung effektiver Pseudonymisierung geleistet.

4.3 Aufbau der Beschreibungssprache `Util`

Die Sprache `Util` erlaubt die nutzbarkeitsorientierte Definition der Anforderungen an eine Pseudonymisierung einer bestimmten Datensammlung. Eine in `Util` formulierte Nutzbarkeitspolitik gilt immer für eine bestimmte Datensammlung und eine geplante Anwendung der zu pseudonymisierenden Daten. Eine solche Politik folgt stets dem *Confidentiality-by-Default*-Prinzips. Es

bezeichnet im Folgenden die Annahme, dass Daten grundsätzlich vertraulich behandelt werden. Die Umsetzung des Prinzips erfolgt durch die Umsetzung von den im Folgenden beschriebenen Regeln.

- In der Politik werden ausschließlich Anforderungen formuliert, die für eine Erfüllung des Verarbeitungszwecks unbedingt erforderlich sind.
- In der Politik werden Klartextdaten anhand ihrer Position in der Datensammlung bzw. den ihnen zugeordneten XML-Tags referenziert. Klartextdaten werden keinesfalls in die Policy aufgenommen.

Hierbei werden zunächst die für die zu erstellenden Pseudonymisierung geplanten Nutzbarkeiten deklariert. Für jede der Nutzbarkeiten werden alle Klartextdaten aus der Datensammlung referenziert, auf denen gemeinsam eine Auswertung der Nutzbarkeit erfolgen soll. Für jede einzelne Nutzbarkeit werden Vertraulichkeitsanforderungen formuliert, welche zu einer Beschränkung der Nutzbarkeit der Pseudonyme im Sinne geltender Datenschutzbestimmungen führen soll.
Im Folgenden bezeichnet $Pol(D)$ eine für eine Datensammlung D in Util formulierte Nutzbarkeitspolitik. D besteht aus semistrukturierten Datensätzen $[d_1,\ldots,d_n]$. Jeder Datensatz D_i besteht aus Daten bzw. Attributwerten $(d_{i,1},\ldots,d_{i,m_i})$.
Mit Util können sowohl einzelne Daten $d_{i,j}$ als auch Datensätze d_i und Teildatensammlungen D_i der Datensammlung D aus einzelnen Elementen oder Datensätzen adressiert werden. Util wurde in XML definiert. Hierbei werden die XML-Elemente und die Baumstruktur zur strukturierten, gleichzeitig in Kombination mit der Wahl der Bezeichner auch menschenlesbaren Darstellung genutzt.

4.3.1 Struktur einer Nutzbarkeitspolitik

Eine Nutzbarkeitspolitik $Pol(D)$ ist datenzentriert aufgebaut. Eine Nutzbarkeitspolitik $Pol(D)$ enthält einen Wurzelknoten <UtilityPolicy>. Dieser wiederum enthält einen Kindknoten <Data>, in dem $Pol(D)$ durch Angabe eines eindeutigen Identifikators mit der adressierten Datensammlung D verknüpft wird. Der Knoten <Data> enthält mindestens einen Kindknoten <DataAttributeSet>. Dieser definiert die Attributwerte aus D, die für die Bereitstellung einer einzelnen Nutzbarkeit erforderlich und dementsprechend in der Pseudonymisierung zu repräsentieren sind. Danach folgt die konkrete Definition der Nutzbarkeitsanforderung in <Utility> als Kindknoten von <DataAttributeSet>. Diese datenzentrierte Adressierung erleichtert die Nachvollziehbarkeit der in eine Pseudonymisierung eingegangenen Daten. Weiterhin erlaubt sie eine übersichtliche Darstellung der Verteilung der Nutzbarkeiten über die Datensammlung und die Nachvollziehung von Daten, für die eine Mehrzahl von Nutzbarkeiten definiert wurden. Jeder Kindknoten <Utility> beschreibt eine Nutzbarkeitsanforderung. In den weiteren Kindknoten des Knotens <Utility> sind alle weiteren, für die Umsetzung der Nutzbarkeit erforderlichen Informationen angegeben. Diese Kindknoten umfassen genau einen <Requirement>-Knoten und mindestens einen <Conditions>- und <Bindings>-Knoten.Im Kindknoten <Requirement> wird die Nutzbarkeit konkretisiert. Die Kindknoten <Conditions> und <Bindings> werden verwendet, um der Nutzbarkeit explizite grundlegende Vertraulichkeitsanforderungen zuzuweisen. Die Deklaration mindestens eines dieser beiden Kind-

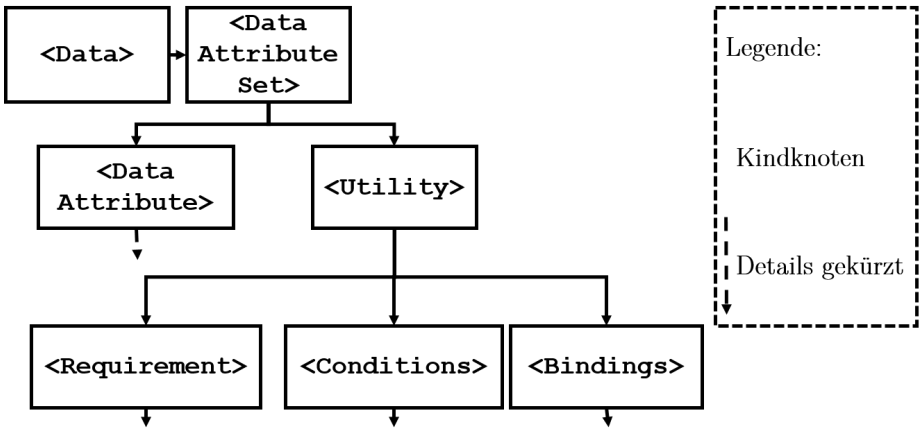

Abbildung 6: *Grundlegende Struktur einer Nutzbarkeitspolitik in* `Util`.

knoten ist zwingend erforderlich. So soll bei der Umsetzung von $Pol(D)$ die Bereitstellung einer Nutzbarkeit ohne die Beschränkung durch weitere technisch-organisatorische Maßnahme verhindert werden. Die grundlegende Struktur einer Nutzbarkeitspolitik ist in Abbildung 6 skizziert.

4.3.2 Datenadressierung

Um Anforderungen an die Nutzbarkeit einer Pseudonymisierung einer gegebenen Datensammlung D in der Beschreibungssprache `Util` formulieren zu können, muss D von `Util` adressiert werden können. Hierzu wird D in ein die Daten beschreibendes `XML`-Format überführt. Dem Elternknoten `<Data>` dieses Formats wird ein einmalig vergebener Wert als Identifikator der Datensammlung zugewiesen. Mit diesem Identifikator kann eine in `Util` formulierte Nutzbarkeitspolitik dieser eindeutig zugeordnet werden. Die Bezeichner der Kindknoten von `<Data>` sind mit den Attributen von D identisch. Dies soll die Adressierung der Datenattribute von D und die Zuordnung der Nutzbarkeitsanforderungen in $Pol(D)$ zu den adressierten Daten ermöglichen.
Für die Zuordnung einer Nutzbarkeitspolitik zu einer Datensammlung wird in der formulierten Nutzbarkeitspolitik $Pol(D)$ unter dem Elternknoten `<Data>` zunächst die zu pseudonymisierende Datensammlung adressiert. Hierfür wird dem Attribut von `<Data>` der Identifikator der Datensammlung als Attributwert zugewiesen. Das Attribut `type` für den Typ der Datensammlung kann mit dem Wert `set` für eine statische, nicht erweiterbare Datensammlung bzw. mit dem Wert `stream` für eine zum Zeitpunkt der Datenpseudonymisierung nicht feste, erweiterbare Datensammlung versehen werden. Hintergrund für die Angabemöglichkeit des Typs der Datensammlung ist, dass in Abhängigkeit des Typs unterschiedlich parametrisierte Pseudonymisierungsverfahren zum Einsatz kommen können. Parameter der Pseudonymisierungsverfahren, die ausschließlich zum Zeitpunkt der Erstellung der Pseudonymisierung erforderlich sind, können bei Datensammlungen vom Typ `set` nach Beendigung der Erstellung dieser verworfen werden. Entsprechende Parameter für Datensammlungen vom Typ `stream` müssen so vorgehalten werden, dass jeder in Zukunft hinzukommende Klartext pseudonymisiert und der Pseudonymisierung geeignet hinzugefügt werden kann.

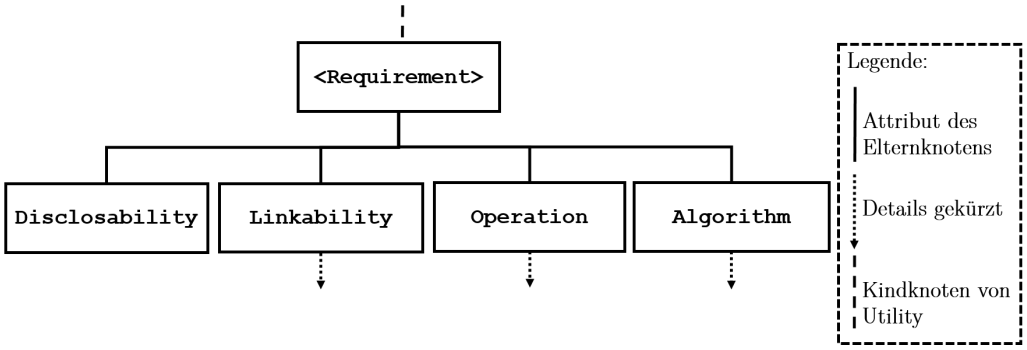

Abbildung 7: *Umsetzung der Anforderungsklassen als* `type`*-Attributwert des* `Requirement`*-Tags in* `Util`*.*

Um für eine Nutzbarkeitsanforderung die zu adressierende Teilmenge D_i der Attribute von D festzulegen, werden die Attribute aus D_i jeweils einem Kindknoten `<DataAttribute>` des `<DataAttributeSet>`-Knotens zugewiesen, der für die Nutzbarkeitsanforderung angelegt wurde. Hierbei wird für jedes Attribut ein Kindknoten `<DataAttribute>` des `<DataAttributeSet>`-Knotens angelegt. Der Bezeichner des betreffenden Attributs wird diesem Knoten als Wert zugewiesen. Diese in einer Nutzbarkeitspolitik verfolgte nutzbarkeitsorientierte Adressierung der Daten soll eine Nachvollziehung der von einer Nutzbarkeit unmittelbar betroffenen Datenattribute erleichtern.

4.3.3 Nutzbarkeitsanforderungen

Eine Nutzbarkeitsanforderung wird im Kindknoten `<Utility>` von `<DataAttributeSet>` definiert. Für die Beschreibung der Nutzbarkeitsanforderung wird der `<Utility>`-Kindknoten `<Requirement>` verwendet. Im Folgenden werden die für die Beschreibung der bisher eingeführten Nutzbarkeitsanforderungen definierten Sprachelemente beschrieben. Die im Folgenden beschriebenen Anforderungen entsprechen den in Kapitel 3 erarbeiteten Anforderungsklassen. Nutzbarkeitsanforderungen werden als Kindknoten-Strukturen des `<Requirement>`-Knotens definiert. Das Attribut `type` des `<Requirement>`-Knotens erhält hierfür einen auf die entsprechende Nutzbarkeit hinweisenden Wert. In Abhängigkeit dieses Werts werden unterschiedliche, an die Erfordernisse der Beschreibung der Nutzbarkeit angepasste Strukturen adressiert. Es werden also in Abhängigkeit von Attributwerten eines Knotens unterschiedliche Kindknotenstrukturen definiert. Diese von einem Attributwert abhängige Definition ist zusätzlich zu der bekannten, auf Elternknoten basierenden Kindknotenstruktur seit `XML 2.0` [134] möglich. Durch die Nutzung dieser Definitionsmöglichkeit wird im Vergleich zur knotenwertbasierten Definition eine Baumebene in der Nutzbarkeitspolitik eingespart. Das Ergebnis ist eine erleichterte Lesbarkeit durch eine übersichtlichere, kompaktere Darstellung.
Im Folgenden wird die Anforderungsformulierung in `Util` für die verschiedenen Anforderungsklassen der Nutzbarkeiten beschrieben. Die Knotenbezeichnungen der einzelnen Nutzbarkeitsanforderungen in `Util` werden in Abbildung 7 zusammengefasst dargestellt. Die Umsetzung der Anforderungsklassen der Nutzbarkeiten Aufdeckbarkeit, Operation und Algorithmus ist in Abbildung 9 schematisch dargestellt. Die schematische Darstellung der Umsetzung der Anforderungsklasse

der Verkettbarkeit bezüglich einer Relation befindet sich in Abbildung 8. Die Umsetzung einzelner konkreter Nutzbarkeitsanforderungen wird exemplarisch dargestellt.

Aufdeckbarkeit Die Nutzbarkeitsanforderung Aufdeckbarkeit wird mit dem type-Attributwert `Disclosability` des Knotens `<Requirement>` deklariert. Weitere die Verfügbarkeit dieser Anforderung einschränkende Bedingungen werden mittels der Strukturen für die Formulierung von Vertraulichkeitsanforderungen formuliert. Diese werden in der Nutzbarkeitspolitik wie für jede Nutzbarkeitsanforderung mittels der Tags `<Conditions>` und `<Bindings>` gesetzt. Dies wird in Abschnitt 4.3.4 beschrieben.

Verkettbarkeit Um die Nutzbarkeitsanforderung Verkettbarkeit zu deklarieren, setzt man den Wert des type-Attributs auf `Linkability`. Die Ausprägung der Verkettbarkeit wird durch eine mehrschichtige Kindknotenstruktur beschrieben. Der Kindknoten `<Relation>` enthält einen Bezeichner für die Relation r, bezüglich derer verkettet werden soll. Beispiele hierfür sind die Gleichheit (Wert `equality`) und die Relation kleiner-gleich bzgl. der zugrundeliegenden Klartextdaten (Wert `less-equal`). Mit dem `<Requirement>`-Kindknoten `<On>` wird definiert, welcher Art die Definitionsmenge der Relation ist. Wenn die Anforderung eine Pseudonym-Pseudonym-Verkettbarkeit ist, wird dies durch die Verwendung des `<On>`-Kindknotens `<Pseudonyms>` kenntlich gemacht. Sollen nur bestimmte Eingabedaten miteinander in Relation gesetzt werden können, so kann dies mit dem Attribut `Position` für die Angabe der Stelligkeit innerhalb der Relation und dem Kindknoten `<SetBySetID>` für die Angabe der Teildatensammlung, aus der die Eingabedaten stammen festgelegt werden. Mit `Position` kann festgelegt werden, an welcher Position in der Relation Elemente der genannten Teilmenge der Eingabedaten stehen können. Indem man zum Beispiel ein Attribut, d.h. eine Spalte einer Datensammlung einer festen Stelligkeit zuweist, kann eine Spalte mit einer anderen bzgl. einer Relation r verglichen werden, nicht aber die Elemente einer Spalte untereinander. Sollen alle Elemente miteinander vergleichbar sein, so wird für jede Position in der Relation derselbe Identifikator gesetzt. Diese Möglichkeit der Unterscheidung erlaubt eine Einschränkung der Verfügbarkeit der Verkettbarkeit auf eine möglichst kleine Eingabemenge.
Für die Pseudonym-Klartext-Verkettbarkeit wird der Kindknoten `<Plaintexts>` verwendet. Die Umsetzung dieser Nutzbarkeit ist in zwei Varianten möglich. Die erste Variante ist die Möglichkeit, alle möglichen Klartextdaten in Relation zu den vorliegenden Pseudonymen zu setzen. Um diese Variante zu deklarieren, wird der Elternknoten `<Plaintexts>` inhaltslos gesetzt. Diese Variante hat zum Nachteil, dass der Klartextraum damit mit einem Known-Plaintext-Angriff aufgespannt werden kann. Details hierzu werden in Kapitel 5.2.1 erläutert. Daher sollte nach Möglichkeit die zweite Variante der Pseudonym-Klartext-Verkettbarkeit genutzt werden. Hierbei werden die zu vergleichenden Klartextdaten im Voraus festgelegt. Um dies in der Nutzbarkeitspolitik anzugeben, wird der Elternknoten `<Plaintexts>` mit einem Kindknoten `<Plaintext>` für jeden zu deklarierenden Klartext definiert. Der Klartext wird als Wert des Kindknotens `<Plaintext>` angegeben. Die Position, an der der Klartext in der Verkettbarkeitsrelation vorkommen kann, wird als Attribut des entsprechenden `<Plaintext>`-Knotens angegeben. Mit dieser Variante der Pseudonym-Klartext-Verkettbarkeit kann die Angriffsfläche reduziert werden. Jedoch geht die Umsetzung dieser

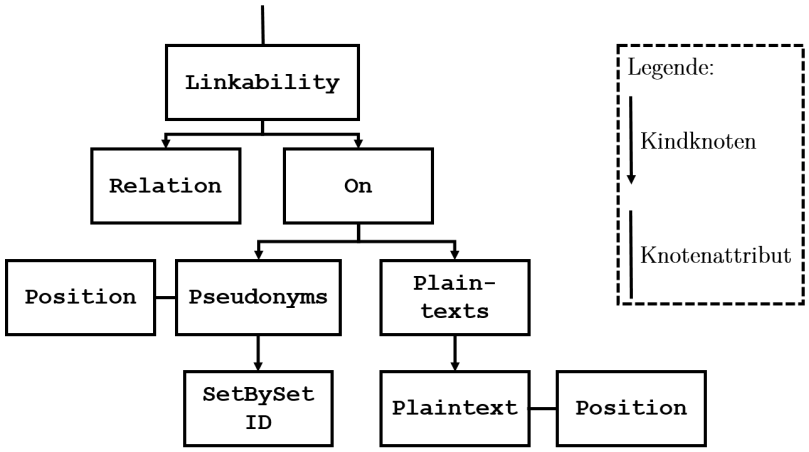

Abbildung 8: *Strukturen der möglichen Ausprägungen des `Requirement`-Typs für die Verkettbarkeit in `Util`.*

Nutzbarkeit wegen der möglichen großen Anzahl der anzugebenden Klartexte mit einem erhöhten Speicherplatzbedarf und einer gewissen Inflexibilität einher.
Sollen verschiedene Relationen auf den Pseudonymen ermöglicht werden, so wird das `<Requirement>`-Element für jede zu definierende Relation einmal mit dem `type`-Wert `Linkability` und den passenden Parametern deklariert. Dies erlaubt auch die Angabe unterschiedlicher Pseudonym-Pseudonym- bzw. Pseudonym-Klartext-Verkettbarkeitsanforderungen. Ein Beispiel ist die Anforderung der Verkettbarkeit bezüglich der Relation Gleichheit. Hier wird der Wert des `<Relation>`-Knotens auf `equality` gesetzt. Weitere Beispiele sind die Verkettbarkeit bezüglich der Relation Kleiner-Gleich (Knotenwert `less-equal`) und der Elementrelation. Bei der Elementrelation wird bestimmt, ob ein gegebenes Datum in einer Menge pseudonymisierter Daten vorkommt. Der Wert des `<Relation>`-Knotens ist `element-of`.
In dieser Dissertation wurde die Verkettbarkeit exemplarisch mit den Relationen Gleichheit, Kleiner-Gleich und Element-von dargestellt. Für die Formulierung einer Nutzbarkeitsanforderung der Verkettbarkeit bezüglich einer weiteren Relation wird der `<Relation>`-Knoten um einen weiteren Knotenwert erweitert. Hier ist zu beachten, dass für die neu hinzugekommene Relation geeignete Pseudonymisierungsverfahren[2] ausgewählt und entsprechende Übersetzungsregeln[3] umgesetzt werden müssen.

Operation Diese Anforderung dient der Umsetzung von Berechnungen, die aus einer einzelnen Operation bestehen. Um die Ausführung einer Operation auf pseudonymisierten Daten zu erlauben, wird dem Attribut `type` des Knotens `<Requirement>` der Wert `Operation` zugewiesen. Im Kindknoten `<Type>` folgt die Konkretisierung der Operation. Exemplarisch sind hier die Werte `Addition` für die Addition und `Multiplication` für die Multiplikation vorgesehen. Sollen verschiedene Operationen auf den Pseudonymen ermöglicht werden, so wird das `<Requirement>`-Element für jede zu definie-

[2] Siehe Kapitel 5
[3] Siehe Kapitel 6.2

rende Operation einmal mit dem type-Wert Operation und den entsprechend passenden Werten für den Kindknoten Type aufgerufen.

Algorithmus Um mehrschrittige Verfahren auf pseudonymisierten Daten zu ermöglichen, wird das Attribut type des Knotens <Requirement> auf den Wert Algorithmus gesetzt. Im Kindknoten <Type> folgt die Konkretisierung des Algorithmus. Algorithmen können unterschiedliche Varianten und Implementationen aufweisen. Ebenso kann die Wahl bestimmter Parameter bei der Initialisierung der Algorithmen eine Rolle spielen. Für einzelne Verfahren existieren Anpassungen, die erst eine Ausführung auf pseudonymisierten Daten erlauben. Ein für pseudonymisierte Daten angepasster Algorithmus wird im Folgenden PET-Variante[4] des Algorithmus genannt. Für unterschiedliche Varianten, Implementationen und Parameter existieren unterschiedlich angepasste Privacy-Varianten. Häufig muss jedoch eine ganz bestimmte Implementierung des Algorithmus eingesetzt werden. Der Einsatz einer bestimmten Implementierung wirkt sich direkt auf die Wahl der Pseudonymisierungsverfahren aus und wird daher als Wert des <Requirement type= "Algorithm">-Kindknotens <Implementation> angegeben.
Varianten eines Algorithmus können sich in der Anzahl der Parameter unterscheiden. Daher kann die Variante eines Algorithmus über die Bezeichner der Parameter als Wert des Kindknotens <ParameterName> des <Parameter>-Knotens angegeben werden. Ist darüber hinaus die Einschränkung auf bestimmte Parameterwerte gewünscht, so werden diese als Werte des <ParameterValue>-Knotens angegeben. Wenn keine Einschränkung gewünscht ist, wird letzterem Knoten kein Wert zugewiesen. Für jeden einzelnen Parameter wird ein <Parameter>-Knoten mit entsprechenden Kindknoten aufgerufen.
Sollen verschiedene Algorithmen auf den Pseudonymen ermöglicht werden, so wird der <Requirement type= "Algorithm">-Knoten für jeden zu definierenden Algorithmus einmal mit den passenden Parametern deklariert. Auch wenn mehr als eine Implementation eines Algorithmus auf den Pseudonymen ermöglicht werden soll, so wird hierfür der <Requirement type= "Algorithm">-Knoten für jede zu definierende Implementation eines Algorithmus' einmal mit den passenden Parametern gesetzt. Für die Angabe unterschiedlicher Implementierungen wird zusätzlich der Kindknoten <Implementation> auf einen Wert gesetzt, der die Implementierung im System referenziert. Soll für eine Variante eines Algorithmus eine beliebige Implementierung genutzt werden können, so kann der Kindknoten <Implementation> ohne Wert deklariert werden.
Ein Beispiel ist die Definierung der Nutzbarkeitsanforderung der in Kapitel 5.4.1 PET-Variante des *k*-Means-Verfahrens. Hier würde der <Type>-Knoten auf den Wert k-means, der <ParameterName>-Knoten auf den Wert k und der <Implementation>-Knoten auf den Wert k-meansMA2020 als Identifikation der Implementierung von [96] gesetzt werden. Um anzugeben, dass die Ausführung des Verfahrens für eine beliebige Clusterzahl möglich ist, würde der <ParameterValue>-Knoten ohne Wert deklariert werden.

[4] PET für Privacy-Enhancing-Technology.

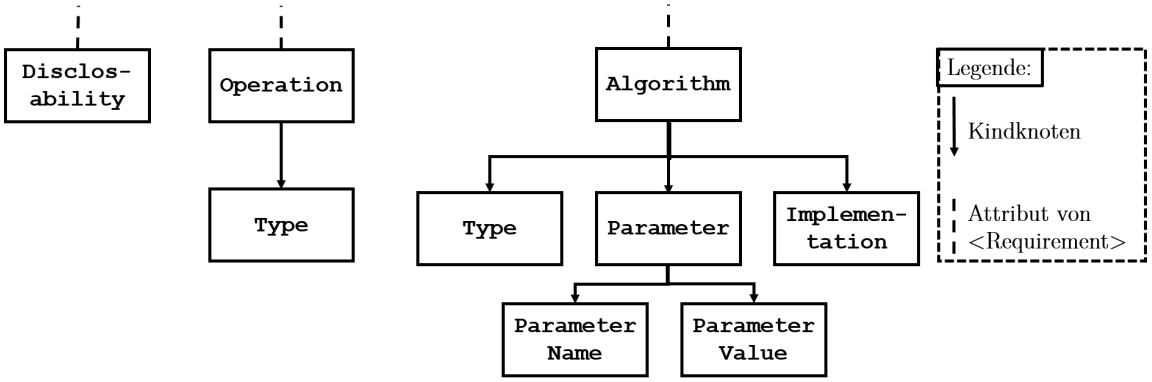

Abbildung 9: *Struktur der* `Requirement`*-Typen für die Aufdeckbarkeit, Operation und Algorithmus in* `Util`*.*

4.3.4 Vertraulichkeitsanforderungen

In Abschnitt 4.3.3 wurde erarbeitet, wie mithilfe der Politik-Sprache `Util` Nutzbarkeitsanforderungen an Pseudonymisierungen formuliert werden. Um die Nutzbarkeiten vor dem Missbrauch durch Dritte zu schützen, wurden in Abschnitt 3.3 Vertraulichkeitsanforderungen formuliert. Diese wurden mittels zusätzlicher technisch-organisatorischer Maßnahmen auf den Pseudonymisierungen umgesetzt. Diese sind vom die Pseudonymisierung verarbeitenden System umzusetzen. Die Umsetzung als Aspekt der mehrschichtigen Sicherheit [80] ist nicht Gegenstand der Betrachtung des Rahmenwerks. Einige Vertraulichkeitsanforderungen können auf den Pseudonymisierungen umgesetzt werden, ohne dass über den Klartext hinausgehende Zusatzinformation erforderlich ist. Daher werden diese Anforderungen lediglich implizit in `Util` abgebildet. Entsprechend sind diese zum Zeitpunkt der Formulierung der Nutzbarkeitspolitik für den Nutzer transparent. Sie werden daher implizite Vertraulichkeitsanforderungen genannt. Für die Umsetzung einiger Vertraulichkeitsanforderungen auf Pseudonymisierungen ist die Angabe bestimmter Zusatzinformation erforderlich. Ohne diese Zusatzinformation ist die Umsetzung der Schutzmaßnahmen auf den Pseudonymisierungen nicht möglich. Für die Formulierung dieser Anforderungen werden daher in `Util` Strukturen bereitgestellt. Sie werden explizite Vertraulichkeitsanforderungen genannt.

Implizite Anforderungen

Unter impliziten Anforderungen werden die Datenminimierung, die Trennung von Pseudonymisierung und Zusatzinformation, die Beachtung des Standes der Wissenschaft und Technik und die Beschränkung der Reidentifizierbarkeit Betroffener zusammengefasst. Diese Anforderungen müssen bei der Umsetzung der Pseudonymisierung berücksichtigt werden. Daher werden sie beim Entwurf von Übersetzungsregeln für die Erstellung von Pseudonymen aus einer `Util`-Politik besonders beachtet. Im Folgenden wird beschrieben, wie sie beim Entwurf von `Util` umgesetzt wurden.

Datenminimierung Die Datenminimierung wird bereits durch die Auswahl der zu pseudonymisierenden Daten und die Einschränkung der Nutzbarkeitsanforderungen auf die für den Anwendungsfall erforderlichen umgesetzt. Eine weitere Maßnahme der Datenminimierung ist, dass in `Util` ausschließlich für die Erstellung der Pseudonymisierung erforderliche Information aus den personenbezogenen Klartextdaten beschrieben wird. Auf Daten, die nicht in die Pseudonymisierung einfließen, wird im Sinne eines Whitelistings[5] in `Util` kein Bezug genommen.

Trennung von Pseudonymisierung und Zusatzinformation Die Trennung von Pseudonymisierung und Zusatzinformation wird umgesetzt, indem `Util` weder pseudonymisierte Daten noch für die Erfüllung der Nutzbarkeitsanforderungen erforderliche Zusatzinformation enthält. Pseudonymisierungen und deren Klartextdaten werden lediglich durch Metainformation referenziert. Ein Beispiel ist die `setID`, mit der unter `<Utility>` eine bestimmte Menge von Klartextdaten referenziert wird. Auf Zusatzinformation wie z.B. kryptographische Schlüssel wird in einer Politik nicht Bezug genommen. Dementsprechend existiert für die Trennungsanforderung keine beschreibende Struktur in `Util`.

Beachtung des Standes der Wissenschaft und Technik Der Stand der Wissenschaft und Technik wird beachtet, indem alle bekannten, das Risiko der Reidentifizierung Betroffener reduzierenden Maßnahmen ergriffen wurden. In `Util` hat der Stand der Wissenschaft und Technik insbesondere die Designentscheidungen beeinflusst. So wurde entschieden, keine Unterscheidung anhand im Klartext vorkommender Daten zuzulassen. Hier hätten sonst Klartextdaten in der Politik aufgezählt werden können. Diese hätten als zusätzliches Hintergrundwissen für einen Angriff auf die Pseudonymisierung genutzt werden können. Auch wurde entschieden, jede Nutzbarkeitsanforderung in einem eigenen `<Utility>`-Knoten zu formulieren und die Vertraulichkeitsanforderungen an jeden einzelnen als Kindknoten zu formulieren. Dies erlaubt es, die Verfügbarkeit der einzelnen Nutzbarkeitsanforderungen mit unterschiedlichen Vertraulichkeitsanforderungen einzuschränken.

Beschränkung der Reidentifizierbarkeit Betroffener Die Einschränkung der Verfügbarkeit der Nutzbarkeitsanforderungen durch die Umsetzung der expliziten Anforderungen hat zur Folge, dass durch die so geschützte Pseudonymisierung die Preisgabe von Information weiter reduziert wird. Dies soll zu einer Erschwerung von Reidentifizierungsangriffen der unintendierten Reidentifizierung Betroffener durch Dritte führen.

Explizite Anforderungen

Zusätzlich zu den impliziten Anforderungen werden in Abschnitt 3.3 Vertraulichkeitsanforderungen formuliert, für deren Umsetzung auf einer Pseudonymisierung die Angabe von Zusatzinformation erforderlich ist. Diese Anforderungen sind die Rollenbindung, die Zweckbindung und die Begrenzung der Speicherdauer.

[5] Siehe z.B. die Erklärungen im Security Insider zu White- und Blacklists: `https://www.security-insider.de/was-ist-eine-whitelist-und-blacklist-a-667574`.

Rollenbindung Im Anwendungsfall können den Nutzern eines Systems unterschiedliche Rollen zugewiesen werden. Diese Rollen erlauben zum Beispiel eine Unterscheidung der Nutzer entsprechend ihrer Vertrauenswürdigkeit oder der von ihnen zu erfüllenden Aufgaben. Diesen Rollen können unterschiedliche Berechtigungen zugewiesen werden. Zum Beispiel kann der Zugriff eines Unternehmensmitarbeiters der Rolle Buchhalter auf die Gehälter der Angestellten im Klartext erlaubt sein. Ein Data Scientist dürfte dies ggf. nicht. Jedoch könnte er befugt sein, mittels Zugriff auf die Operation der Summenberechnung Mittelwertberechnungen durchzuführen und so Rückschlüsse auf die Durchschnittsgehälter bestimmter Berufsgruppen zu erhalten.
Um die Verfügbarkeit einer Nutzbarkeit an eine bestimmte Rolle zu binden, wird der `<Utility>`-Kindknoten `<Bindings>` mit dem Kindknoten `<Binding>` aufgerufen und das `<Binding>`-Attribut `type` wird mit dem Wert `role` belegt. Dieser Attributwert erlaubt dann die Deklarierung des Kindknotens `<Role>`. Der Bezeichner der Rolle wird dem `<Role>`-Knoten zugewiesen. Um die Verfügbarkeit einer Nutzbarkeit für mehrere Rollen zu ermöglichen, wird für jede einzelne Rolle ein `<Binding>`-Knoten aufgerufen und mit einem zur entsprechenden Rolle passenden Wert für den `<Role>`-Knoten belegt. Wird mehr als eine Rollenbindung deklariert, so wird die entsprechende Nutzbarkeit den Rollen unabhängig voneinander ermöglicht.

Zweckbindung Um die zweckgebundene Verarbeitung einer Nutzbarkeit eines Pseudonyms festzulegen, wird die Verfügbarkeit dieser Nutzbarkeit an den Zweck der Datenverarbeitung gebunden. Hierfür wird der `<Utility>`-Kindknoten `<Bindings>` mit dem Kindknoten `<Binding>` aufgerufen und das `<Binding>`-Attribut `type` wird mit dem Wert `purpose` belegt. Mit diesem Attributwert ist die Deklarierung des Kindknotens `<Purpose>` möglich. Der Wert des `<Purpose>`-Knotens entspricht dem Bezeichner des intendierten Zwecks der Verarbeitung. Auch bei dieser Variante der Bindung der Verfügbarkeit der Nutzbarkeit ist die Angabe mehrerer Zwecke durch mehrfachen Aufruf des `<Binding>`-Knotens möglich. Jeder einzelne Zweck wird durch Angabe des entsprechenden Bezeichners als Wert des `<Purpose>`-Knotens deklariert.

Zusammenfassend können beide Arten der Bindung für eine Nutzbarkeit kombiniert werden. Dies erfolgt durch die mehrfache Nutzung des `<Binding>`-Knotens mit entsprechender Belegung des `type`-Attributs mit den Werten `Role` bzw. `Purpose`. Es wird angenommen, dass das Vorliegen eines Verarbeitungszwecks oder die Erfüllung einer Rolle ein Subjekt hinreichend für die Auswertung einer Nutzbarkeit autorisiert. Die Erlaubnis zur Verarbeitung einer Nutzbarkeit durch ein Subjekt in einer bestimmten Rolle geschieht daher unabhängig von der Erfüllung eines bestimmten Verarbeitungszwecks. Die Baumstruktur von `Bindings` und `Conditions` ist in Abbildung 10 skizziert.

Begrenzung der Speicherdauer Die Begrenzung der Speicherdauer ermöglicht es, vor Freigabe einer Nutzbarkeit die Systemzeit gegen ein in der Politik anzugebendes Datum abzugleichen. Somit kann zum Beispiel ein Löschmechanismus implementiert werden, der nach Verstreichen des angegebenen Datums die entsprechende Nutzbarkeit löscht. So wird sichergestellt, dass die Daten nach Verstreichen des angegebenen Datums nicht mehr verarbeitet werden. Das

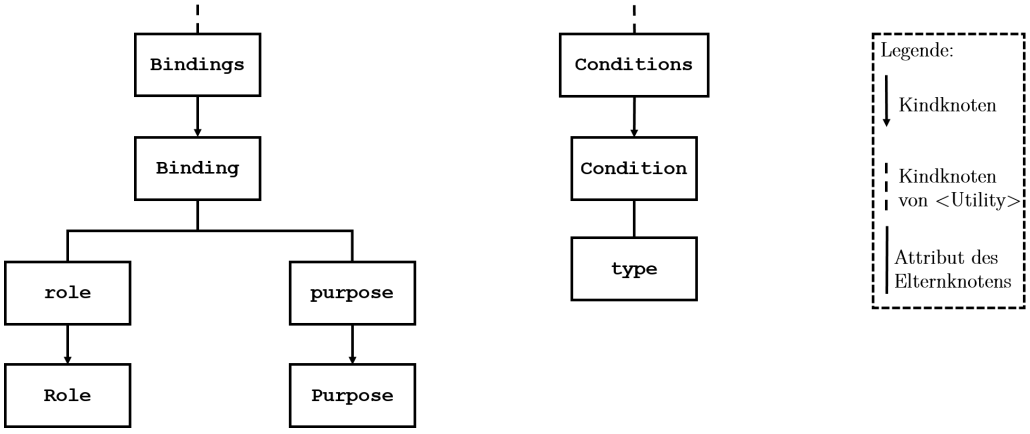

Abbildung 10: *Die Strukturen der* `<Utility>-Kindknoten` `<Bindings>` *und* `<Conditions>`.

Ablaufdatum kann als negative Bedingung aufgefasst werden. Um dies in einer `Util`-Politik umzusetzen, wird der `<Utility>`-Kindknoten `<Conditions>` mit dem Kindknoten `<Condition>` aufgerufen. Das `<Condition>`-Attribut `type` wird mit dem Wert `expirationTime` belegt. Der Wert des `<Condition>`-Knotens entspricht dann der Systemzeit, zu der der Zugriff auf die Nutzbarkeit letztmalig möglich ist.
Auf diese Weise ist durch die Erweiterung der Menge der möglichen Werte des `<Condition>`-Attributs `type` die Hinzunahme weiterer, die Verfügbarkeit einer Nutzbarkeit einschränkender Bedingungen möglich. In dieser Arbeit wird dies als zukünftige, im Rahmen expliziter Anwendungsfälle ermöglichte Option betrachtet. Um die Konkretisierungen dieser weiteren Bedingungen in `Util` umzusetzen, muss die Syntax um entsprechende Werte für das `type`-Attribut ergänzt werden.

In den vorangegangenen Abschnitten wurden die Struktur und die Elemente der Politik-Sprache `Util` erarbeitet. Beispiele für die Instanziierung von Nutzbarkeitspolitiken in `Util` werden im Anhang 8.3 gelistet.

4.4 Designentscheidungen

Motivation der Entwicklung von `Util` ist die Ermöglichung der Beschreibung von Nutzbarkeits- und Vertraulichkeitsanforderungen an eine Pseudonymisierung in Form einer Nutzbarkeitspolitik. Ziel ist die Gestaltung dieser Beschreibungsmöglichkeit derart, dass aus einer Nutzbarkeitspolitik automatisiert nutzbarkeitserhaltende Pseudonymisierungen generiert werden können. Ein nachgeordnetes Ziel ist die Ermöglichung einer Nachvollziehbarkeit der von den Nutzbarkeitsanforderungen adressierten Klartextdaten. Ein weiteres nachgeordnetes Ziel ist die Berücksichtigung von Privacy-by-Design. Zur Erreichung dieses Ziels werden Designentscheidungen getroffen. Diese werden im Folgenden dargestellt und in Tabelle 2 zusammengefasst.

Tabelle 2: *Zusammenfassung der Designentscheidungen und ihrer Umsetzung.*

Ziel	Designentscheidung
Automatisierte Ableitbarkeit einer Pseudonymisierung	Maschinenlesbarkeit Baumstruktur (`XML`)
Nachvollziehbarkeit der von Nutzbarkeitsanforderungen adressierten Klartextdaten	Anordnung der Anforderungen entlang der involvierten Klartextdaten
Privacy-by-Design	- Datenminimierung - Confidentiality-by-Default - keine Klartextdaten in Politik - keine Pseudonymisierung in Abhängigkeit von Klartextwerten

4.4.1 Automatisierte Ableitbarkeit einer Pseudonymisierung

Um aus einer Nutzbarkeitspolitik automatisiert eine nutzbarkeitserhaltende Pseudonymisierung erstellen zu können, wird `Util` maschinenlesbar entwickelt. Um die unterschiedlichen Anforderungen in einer Politik abbilden zu können, ist eine Struktur erforderlich, die leicht traversierbar ist und unterschiedliche Ebenen mit Fallunterscheidungen erkennen lässt. Eine Baumstruktur erfüllt diese Anforderungen. Die Markup-Syntax von `XML` erlaubt die Definition von maschinenlesbaren Baumstrukturen. Diese sind als Wurzelknoten mit Kindknoten bis hin zu Blättern definiert. Wurzel- und Kindknoten können mit Attributen versehen werden. In Abhängigkeit von den Attributwerten können unterschiedliche Kindknoten-Strukturen ausgewählt werden. Die Blätter beinhalten Daten, die vordefiniert festgelegt werden können. `XML` ermöglicht so die präzise Adressierung der Elemente, die für die Anforderungsbeschreibung in einer Nutzbarkeitspolitik erforderlich sind. Die hierarchische Baumstruktur der Knoten und die Sichtbarkeit der Knotenbezeichner ermöglichen eine strukturierte menschenlesbare Darstellung der Nutzbarkeitsanforderungen. Die Darstellung komplex zusammenhängender Nutzbarkeits- und Vertraulichkeitsanforderungen wird so erleichtert. Aus diesen Gründen wird `Util` in `XML` definiert.

4.4.2 Nachvollziehbarkeit der von Nutzbarkeitsanforderungen adressierten Klartextdaten

Um die Klartextdaten, die in einer Nutzbarkeitspolitik adressiert werden möglichst übersichtlich darzustellen, wird `Util` so konzipiert, dass eine Nutzbarkeitspolitik für eine bestimmte Datensammlung D definiert wird. D bildet eine Obermenge aller in der Nutzbarkeitspolitik adressierten Klartextdaten d. Für jede Nutzbarkeitsanforderung wird zunächst eine Teilmenge D_i von D als `<DataAttributeSet>`-Knoten definiert, die diese Anforderung erfüllen soll. An diese Teilmenge werden als Kindknoten die Ausprägung der Nutzbarkeitsanforderung und die zugehörigen Vertraulichkeitsanforderungen formuliert. Ergebnis dieser Designentscheidung ist, dass zum einen auf einer Ebene unter dem Wurzelknoten die adressierte Datensammlung ersichtlich ist. Zum anderen ist zwei Ebenen tiefer die Partitionierung der Datensammlung in Elemente der Potenzmenge der Datensammlung in Abhängigkeit der gewählten Nutzbarkeitsanforderungen ersichtlich. So kann zum Beispiel nachvollzogen werden, welcher Klartext besonders viele Nutzbarkeitsanforde-

rungen aufweist. Dies soll die Durchführung von Risiko- und Datenschutzfolgenabschätzungen unterstützen.

4.4.3 Privacy-by-Design

Um das Prinzip der Privacy-by-Design auch beim Entwurf von Nutzbarkeitspolitiken zu berücksichtigen, werden die folgenden Prinzipien betrachtet und angenommen.

Datenminimierung Anwender formulieren in der Nutzbarkeitspolitik ausschließlich für den Anwendungsfall notwendige Nutzbarkeitsanforderungen. Vertraulichkeitsanforderungen sind zwingend anzugeben. So kann der Zugriff auf die Nutzbarkeitsanforderungen möglichst passgenau eingeschränkt werden.

Confidentiality-by-Default Für Klartextdaten d aus D, die im Anwendungsfall nicht genutzt werden, werden in der Nutzbarkeitspolitik keine Anforderungen formuliert. Dementsprechend werden sie in der Pseudonymisierung nicht repräsentiert und bleiben so vertraulich. Eine weitere Entscheidung ist, Werte von Klartextdaten nicht explizit in einer Nutzbarkeitspolitik zu beschreiben. Vielmehr werden diese Werte anhand ihrer Position in D referenziert. So soll verhindert werden, dass Klartextdaten aus der Nutzbarkeitspolitik ausgelesen werden können. Dementsprechend ist es zum Beispiel nicht möglich, in Abhängigkeit der Klartextwerte mit unterschiedlichen Anforderungen zu pseudonymisieren. Damit soll verhindert werden, dass eine Kenntnisnahme einzelner Klartext-Attributwerte aus D zur Inferierung der Klartexte pseudonymisierter Attributwerte verwendet werden können.

4.5 Fazit zur Beschreibungssprache `Util`

Für den Entwurf einer Politik-Sprache für Nutzbarkeitspolitiken wurden zu Beginn des Kapitels fünf Anforderungen formuliert. Im Folgenden wird zusammenfassend dargestellt, wie diese Anforderungen bei der Konzeption von `Util` adressiert wurden.

Formulierbarkeit der Anforderungen Für jede der in Kapitel 3 eingeführten Anforderungsklassen werden in `Util` Strukturen zur Formulierung der Anforderungen definiert. Für einige dieser Anforderungsklassen wurden Strukturen für die Formulierung konkreter beispielhafter Ausprägungen definiert.

Maschinenlesbarkeit Um die Nutzbarkeitspolitiken maschinenlesbar formulieren zu können, wird `Util` in `XML` der Version `2.0` definiert. Ein Vorteil der Nutzung von `XML` ist, dass verschiedene Parser, Interpreter und APIs für die unterschiedlichen Programmiersprachen zur Verfügung

stehen [134]. Somit kann `Util` erleichtert in unterschiedlichen Kontexten und Anwendungsfällen umgesetzt werden.

Menschenlesbarkeit Damit die Nutzbarkeitspolitiken auch von Menschen gelesen und nachvollzogen werden können, wurden für die Elemente der Sprache selbsterklärende Bezeichnungen in englischer Sprache gewählt. Es wurden Elemente der `XML`-Spezifikation `2.0` genutzt, um eine kompakte und damit im Rahmen der technischen Möglichkeiten übersichtlichere Darstellung der Politiken zu ermöglichen. Ein weiterer Vorteil von XML, der die Lesbarkeit durch Menschen erhöhen kann, ist die Baumstruktur. Eine einzelne Nutzbarkeitsanforderung mit den zugehörigen Vertraulichkeitsanforderungen entspricht einem Teilbaum in der Gesamtstruktur einer Politik. Es bleibt jedoch anzumerken, dass die Menschenlesbarkeit ein nachgeordnetes Ziel der Konzeption ist. Für eine stark vereinfachte Lesbarkeit kann z.B. eine grafische Nutzeroberfläche für die Darstellung der Elemente von `Util` genutzt werden. Ein Beispiel einer solchen Oberfläche ist in Abbildung 5 des Anhangs zu finden. Gleiches gilt für eine vereinfachte Eingabemöglichkeit zur Definierung einer einzelnen Nutzbarkeitspolitik. Beispiele für die Formulierung von Nutzbarkeitspolitiken in `Util` sind im Anhang in Abschnitt 8.3 gelistet.

Dokumentierbarkeit Durch die Umsetzung der expliziten Vertraulichkeitsanforderungen als Elemente einer Nutzbarkeitspolitik dient diese als Dokumentation der auf diese Weise umgesetzten Datenschutzprinzipien.

Privacy-by-Design Durch die Umsetzung der Datenminimierung und des Prinzips der Confidentiality-by-Default wird Privacy beim Entwurf der Sprache berücksichtigt. Die Nutzbarkeitspolitik enthält keine Information, die nicht auch in der zugehörigen Pseudonymisierung umgesetzt ist. Ein System, das Zugriff auf die Pseudonymisierung in ihrer Gesamtheit hat, erfährt durch Kenntnisnahme der Politik keinen Informationsgewinn. Ein System, das keinen Zugriff auf die Pseudonymisierung hat, jedoch Kenntnis über die Politik erhält, gewinnt durch die Adressierung der Daten lediglich eine teilweise Kenntnis der Struktur des Datensatzes im Klartext. Über die Pseudonymisierung erlangt es Kenntnis der Nutzbarkeitsanforderungen, jedoch ohne diese auswerten zu können. Insgesamt ist diese Information in einem Umfang gegeben, der höchstens dem entspricht, der dem Datenverarbeitenden durch die Pseudonymisierung vorliegt.

5 Nutzbarkeitserhaltende Pseudonymisierungsverfahren

In dieser Arbeit werden Nutzbarkeitsanforderungen in Pseudonymisierungen umgesetzt. Zur Erstellung von nutzbarkeitserhaltenden Pseudonymisierungen sind geeignete Pseudonymisierungsverfahren erforderlich. In der Literatur ist eine Vielzahl von Pseudonymisierungsverfahren bekannt. In diesem Kapitel werden für die in Kapitel 3 beschriebenen Anforderungsklassen der Nutzbarkeiten Pseudonymisierungsverfahren beschrieben, die bei der Umsetzung nutzbarkeitserhaltender Pseudonymisierung im erarbeiteten Rahmenwerk verwendet werden. Hierbei werden bekannte kryptographische Verfahren aufgegriffen und für die Konstruierung der nutzbarkeitserhaltenden Pseudonymisierungsverfahren verwendet. Für die Nutzbarkeitsanforderungen *Verkettbarkeit bezüglich der Relation Element-von* und *Algorithmus k-Means* werden eigene Verfahren vorgeschlagen und beschrieben.

Pseudonymisierungsverfahren überführen die einzelnen Klartextdaten in eine pseudonymisierte Form, die den Nutzbarkeitsanforderungen genügt. Eine Pseudonymisierung ist die Gesamtheit der durch die Anwendung eines Pseudonymisierungsverfahrens auf eine Datensammlung erzeugten Pseudonyme. Pseudonyme können mit unterschiedlichen Verfahren erzeugt werden. Dies umfasst Hash- und Verschlüsselungsverfahren [5]. Möglich sind auch Ersetzungsverfahren, die Klartextdaten durch fest vorgegebene Pseudonyme ersetzen [136].

Dem Stand der Wissenschaft entsprechende Verschlüsselungsverfahren und einige Hash-Verfahren erfordern die Nutzung von sogenannter Zusatzinformation zur sicheren Verarbeitung. Je nach Verfahren können dies öffentliche und private oder symmetrische Schlüssel oder Salts sein. Im Gegensatz zu Hash-Verfahren existieren zu modernen Verschlüsselungsverfahren Entschlüsselungsmechanismen, die mittels Nutzung von geheimzuhaltenden Schlüsseln eine Entschlüsselung der Chiffrate erlauben. Werden Pseudonymisierungen unter Nutzung solcher zusätzlicher Daten generiert, so ist zum Schutz der Vertraulichkeit der Klartextdaten die Separierung dieser zusätzlichen Daten von den Pseudonymisierungen erforderlich [136].

Bei der Verarbeitung von nutzbarkeitserhaltenden Pseudonymisierungen ist in vielen Fällen der Zugriff auf Schlüssel und Salts erforderlich. Zum Schutz der Vertraulichkeit der Klartextdaten ist der Zugriff auf die Schlüssel und Salts zusätzlich zu schützen. Im die Pseudonymisierung verarbeitenden System müssen daher Maßnahmen zur Überwachung der Verarbeitung der Pseudonymisierungen implementiert sein.

In dieser Arbeit wird beschrieben, wie ausgehend von einer maschinenlesbaren Anforderungsbeschreibung automatisiert Pseudonymisierungen erstellt werden können. Hierfür ist die Auswahl von geeigneten Pseudonymisierungsverfahren erforderlich. Im vorliegenden Kapitel werden daher

nutzbarkeitserhaltende Pseudonymisierungsverfahren beschrieben. Diese Verfahren sind bereits in der Literatur bekannt. Als Baustein für die Erarbeitung der Übersetzungsregeln in Kapitel 6 werden die Verfahren nach den zu erfüllenden Nutzbarkeitsanforderungen systematisiert.
Nutzbarkeitserhaltende Pseudonymisierungsverfahren können in zwei Kategorien eingeteilt werden: Auf kryptographischen Verfahren basierende und auf Informationsreduktion basierende Pseudonymisierungsverfahren. Bei auf kryptographischen Verfahren basierenden Pseudonymisierungsverfahren werden die Pseudonyme durch die Verschlüsselung der zugrundeliegenden Klartextdaten erzeugt. Typischerweise wird kryptographische Zusatzinformation verwendet. Dies können Schlüsselpaare, geheime Schlüssel und Salt-Werte sein. Die Nutzbarkeit eines Pseudonyms kann die Nutzung und somit Kenntnis der Zusatzinformation erfordern. Beispiele sind homomorphe Verschlüsselungsverfahren, ordnungserhaltende Verfahren und kryptographische Hash-Funktionen. Für diese Verfahren ist die Bereitstellung von Zusatzinformation in Form von kryptographischen Schlüsseln und Salts erforderlich.
Bei auf Informationsreduktion basierenden Verfahren werden die Pseudonyme durch die Anwendung von informationsreduzierenden, eine gewisse Nutzbarkeit erhaltenden Verfahren auf den Klartextdaten erzeugt. Hierbei werden Pseudonyme erzeugt, die jene Eigenschaften erhalten, die eine Verarbeitung der Nutzbarkeit ermöglichen. Diese Erhaltung der Eigenschaften der Klartextdaten spiegelt sich in der Syntax der Pseudonyme wider. Beispiele hierfür sind Verfahren, die die Länge der Klartextdaten in den Pseudonymen erhalten und Verfahren, die einen Präfix oder Suffix fester Länge aus dem Klartext in das Pseudonym überführen [56]. Für Verfahren, die nicht auf sicheren kryptographischen Primitiven basieren, sind häufig angreifbar. Ein Beispiel für Angriffe auf Verfahren, die den in [56] veröffentlichte Verfahren strukturell ähnlich sind, ist [31].

In dem vorliegenden Kapitel werden für die in Kapitel 3 erarbeiteten Nutzbarkeitsanforderungen geeignete Pseudonymisierungsverfahren vorgestellt. Für jede der Anforderungen existieren Anwendungsfälle, die in der Literatur beschrieben sind. Diese werden zur Motivierung der ausgewählten den Pseudonymisierungsverfahren zugrundeliegenden Techniken angeführt. Es ist zu beachten, dass die referenzierten Arbeiten bei der Formulierung von Anforderungen und der Beschreibung von Pseudonymisierungsverfahren keiner einheitlichen Terminologie folgen. In der vorliegenden Arbeit werden die Begrifflichkeiten vereinheitlicht. Dies wird umgesetzt, indem die Verfahren als Pseudonymisierungsverfahren bezeichnet werden, deren Ziel die Erfüllung der Definition der Pseudonymisierung nach Definition 10 durch Anwendung von Datenpseudonymisierung nach Definition 9 ist. Entsprechend dem Ziel der vorliegenden Arbeit werden die Pseudonymisierungsverfahren im vorliegenden Kapitel nach den in Kapitel 3 eingeführten Anforderungsklassen systematisiert.
Zunächst werden im Folgenden Vorbemerkungen zur Sicherheit von Pseudonymisierungsverfahren und der Auswertung von Nutzbarkeiten ausgeführt. Davon ausgehend werden in den Kapiteln 5.1, 5.2, 5.3 und 5.4 Pseudonymisierungsverfahren für die erarbeiteten Nutzbarkeitsanforderungen der Anforderungsklassen aus Kapitel 3 beschrieben. Schließlich werden die im Kapitel erarbeiteten Erkenntnisse zu nutzbarkeitserhaltenden Pseudonymisierungsverfahren in Abschnitt 5.5 inhaltlich zusammengefasst.

Vorbemerkungen zur Sicherheit von Pseudonymisierungsverfahren

Pseudonymisierungsverfahren, die dedizierte Eigenschaften der Klartextdaten erhalten, sind in der Forschung wohlbekannt. Meist werden für einen bestimmten Anwendungsfall Verfahren konstruiert, die für den Einsatzbereich nutzbare Pseudonyme aufbereiten. Jedoch schützen diese Verfahren nicht immer zuverlässig die Vertraulichkeit der Daten. Hauptsächlicher Grund ist, dass die Verfahren auf Techniken basieren, die nicht dem Stand der Wissenschaft in der modernen Kryptographie entsprechen. Vielmehr werden Verfahren entwickelt, die in erster Linie eine bestimmte Nutzbarkeit erhalten sollen. Gleichzeitig sollen die Daten nicht im Klartext vorliegen. Das Ergebnis sind Pseudonymisierungsverfahren, die auf leicht anzugreifenden Verschleierungstechniken basieren. Diese Techniken lassen sich im Wesentlichen auf die folgenden Verfahrensarten zurückführen: die Verschiebung der Klartextdaten durch Addieren oder Anmultiplizieren von Konstanten [117], die Anwendung von linearen Funktionen auf die Klartextdaten [117] oder die Anwendung von homomorphen [154] oder eigenschaftenerhaltenden Verschlüsselungsverfahren [147], die nicht den bekannten Sicherheitsstandards entsprechen bzw. bereits gebrochen sind. Gerade beim Vorliegen mehrerer Pseudonyme kann ein Klartextdatum durch Aufstellen linearer Gleichungssysteme leicht herausgerechnet werden.
Ein Beispiel für das Anwenden linearer Funktionen und das Addieren von Konstanten ist die Aufbereitung von Klartextdaten mit einem Verschleierungsverfahren in einer Weise, die die Ausführung des k-Means-Clustering-Verfahrens auf den Daten erlaubt [117]. Ein Beispiel für den Einsatz von unsicheren Verschlüsselungsverfahren sind symmetrische homomorphe Verschlüsselungsverfahren. Beispiele für bereits gebrochene Verfahren sind [93] und [152]. Ein weiteres Beispiel ist der Einsatz von nicht sicheren ordnungserhaltenden Verschlüsselungsverfahren wie [3].
Um das Risiko der Reidentifizierung Betroffener durch die Auswertung der Nutzbarkeiten zur Inferierung der Klartextdaten zu reduzieren, gleichzeitig aber die erforderliche Nutzbarkeit bereitzustellen, sollen Pseudonymisierungsverfahren möglichst auf dem Stand der Wissenschaft genügenden Verschlüsselungsverfahren basierend entworfen werden. Eine weitere Anforderung ist, nach Möglichkeit die Pseudonymisierung durch weitere Maßnahmen zu schützen. Dies können Maßnahmen sein, die im Pseudonymisierungsverfahren als sogenannte Härtung ergänzt werden [5].
Ein Beispiel für das Härten eines Pseudonymisierungsverfahrens ist das gezielte, an die zu erwartenden Daten angepasste Einfügen von Dummy-Werten in einen Bloom-Filter. Der Bloom-Filter soll es erlauben, in einer ansonsten vertraulich gehaltenen Menge nach bestimmten Elementen zu suchen. Um zu verhindern, dass dabei nach und nach ein Aufzählen der Elemente der Menge möglich wird, wird die Unsicherheit des Bloom-Filters durch das Einkodieren von Dummy-Werten zusätzlich erhöht [7]. Dabei muss beachtet werden, dass die Falsch-Positiv-Rate des Bloom-Filters durch das Einfügen der Dummy-Werte möglicherweise ansteigt [7].
Ein weiteres Beispiel für Maßnahmen, die über die Konstruktion des Pseudonymisierungsverfahrens hinausgehen, ist die zusätzliche Verschlüsselung des nutzbarkeitserhaltenden Pseudonyms mit einem randomisierten Verschlüsselungsverfahren [123, 133]. Hierbei wird das Risiko der Aufdeckung der Klartextdaten durch die Einschränkung der Verfügbarkeit der Nutzbarkeit weiter reduziert. Diese Maßnahme wird in dieser Arbeit im Rahmen des Bindens des Pseudonyms an Rollen bzw. Zwecke umgesetzt.

Nutzbarkeitserhaltende Pseudonymisierungen erlauben per Definition einen intendierten Informationsabfluss. Zusätzlich zu diesem kann weiterer, nicht unbedingt gewollter Abfluss von Information nicht immer verhindert werden. Daher ist die Einschränkung der Verfügbarkeit der Pseudonymisierungen durch die Umsetzung von Vertraulichkeitsanforderungen unerlässlich. Ein Beispiel ist die Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Gleichheit als intendierter Informationsabfluss. Sind Pseudonyme bezüglich Gleichheit verkettbar, so kann gezählt werden, wie häufig einzelne Pseudonyme in einer Datensammlung vorkommen. Die Ergebnisse der Häufigkeitsanalyse [85] können direkt auf die unterschiedlichen zugrundeliegenden Klartextdaten übertragen werden. Daraus kann weitere Information über die Klartextdaten inferiert werden. Ist zum Beispiel bekannt, dass die Daten Information über das Geschlecht Betroffener enthalten, so kann aus der Verteilung der unterschiedlichen Pseudonyme mit großer Wahrscheinlichkeit auf das Geschlecht der Betroffenen geschlossen werden[1].

Vorbemerkungen zur Auswertung von Nutzbarkeiten

Die Auswertung der Nutzbarkeiten von Pseudonymisierungen erfolgt für die meisten Anforderungsklassen im Rahmen von Mehrparteienprotokollen. Vor der Auswertung der Nutzbarkeit wird die Erfüllung der Vertraulichkeitsanforderungen überprüft. Die Rolle des Analyst umfasst alle Rollen, die die Vertraulichkeitsanforderungen einer Nutzbarkeitsanforderungen so erfüllen, dass sie zur Auswertung der Nutzbarkeit berechtigt sind. Eine weitere Rolle mit grundlegenden Aufgaben ist der Dekryptor. Der Analyst hat Zugriff auf die Pseudonyme und führt im Rahmen der Auswertung der Nutzbarkeiten Berechnungen auf diesen durch. Ist für die Auswertung der Nutzbarkeit eine Entschlüsselung erforderlich, so stellt der Analyst eine Entschlüsselungsanfrage an den Dekryptor. Dieser hat Zugriff auf die Schlüssel und entschlüsselt nach Prüfung der Zulässigkeit der Anfrage das Berechnungsergebnis. Die Prüfung der Zulässigkeit erfolgt über technisch-organisatorische Maßnahmen, die auf dem verarbeitenden System passgenau auf die auszuwertende Nutzbarkeit abgestimmt werden müssen. Durch die Interaktion voneinander unabhängiger Parteien können die Pseudonymisierungen und Zusatzinformation wie Schlüsselmaterial auf entsprechend separaten Systemen gespeichert werden. Damit wird die Forderung der Datentrennung umgesetzt.
Für die Auswertung der Nutzbarkeitsanforderungen der Verkettbarkeit bezüglich Gleichheit und der Relation Kleiner-Gleich benötigt der Analyst konstruktionsbedingt nach aktuellem Stand der Wissenschaft keine Zusatzinformation. Daher ist nach Erstellung und Erhalt der Pseudonymisierung keine Interaktion mit dem Dekryptor erforderlich. Alle anderen vorgestellten Nutzbarkeitsanforderungen basieren auf Verfahren, die die Entschlüsselung eines Berechnungsergebnisses erfordern. Sie können nur in Interaktion mit dem Dekryptor ausgewertet werden. Für eine Konstruktion ohne Interaktion müssten Verfahren gewählt werden, die im Vergleich zu den gewählten Verfahren angreifbar sind. Dementsprechend wäre die Risikominderung der Pseudonymisierungen nach dem Stand der Wissenschaft nicht mehr gegeben. Ein Beispiel ist das Ersetzen des in Abschnitt 5.2.3 beschriebenen Verfahrens durch ein Verfahren, das auf dem Erzeugen von Pseudonymen durch Anwenden einer deterministischen Hash-Funktion basiert.

[1] siehe z.B. die Ergebnisse in [50] zu Reidentifizierungsangriffen, bei denen eine Häufigkeitsanalyse zum Einsatz kommt.

Im Vergleich zum hier gewählten Verfahren würde der zusätzliche Informationsabfluss über die zugrundeliegenden Klartextdaten größer sein. Es könnte eine Häufigkeitsanalyse und ein Wörterbuchangriff durchgeführt werden. Die vorgestellte Alternative bietet hier einen größeren Schutz der Vertraulichkeit und ist im Vergleich zu weiteren Schutzmechanismen wie der sicheren Mehrparteienberechnung [71] praktisch effizient. Zusätzlich erfordert sie keine Aufteilung der Pseudonyme auf mehrere Parteien.
In der vorliegenden Arbeit wird vorausgesetzt, dass die Systeme, die eine Pseudonymisierung verarbeiten am Schutz der Vertraulichkeit der Klartextdaten interessiert sind. Insbesondere wird vorausgesetzt, dass Analyst und Dekryptor im Sinne des Honest-but-Curious-Angreifermodells [122] das Protokoll einhalten und nicht zur unerlaubten Aufdeckung der Klartextdaten kollaborieren. In der Praxis kann dies durch geeignete Monitoring-Systeme sichergestellt werden. Monitoring-Systeme werden in der IT-Sicherheit zum Beispiel in der Überwachung der Kommunikation in Netzwerken verwendet [70]. Es wird daher angenommen, dass der Einsatz solcher Systeme zur Überwachung der protokollkonformen Verarbeitung von Pseudonymen in der Praxis sinnvoll ist.

Im Folgenden werden für die in der Dissertation erarbeiteten Anforderungsklassen und exemplarisch vorgestellten Nutzbarkeitsanforderungen geeignete Pseudonymisierungsverfahren beschrieben.

5.1 Anforderungsklasse Aufdeckbarkeit

Um aufdeckbare Pseudonyme zu erzeugen, werden drei Verfahrensarten unterschieden. So existieren Verfahren, die auf Secret-Sharing basieren [21]. Andere Verfahren basieren auf probabilistischen symmetrischen [62, 123] oder asymmetrischen [84] Verschlüsselungsverfahren.
Im Falle der Secret-Sharing-Verfahren wird ein Klartextdatum in eine Menge von sog. Shares überführt. Um das den Shares zugrundeliegende Klartextdatum rekonstruieren zu können, müssen alle oder eine Teilmenge festgelegter Größe der Shares vorliegen und gemäß dem der Konstruktion entsprechenden Interpolationsverfahren zum Klartextdatum zusammengesetzt werden [139].
Bei der Verwendung von symmetrischen Verschlüsselungsverfahren ist für die Aufdeckung des Klartextdatums Kenntnis des geheimen Schlüssels erforderlich. Der Advanced-Encryption-Standard AES-256 [45, 32] ist ohne die Ergreifung weiterer Maßnahmen deterministisch. Dies ermöglicht die Verkettung der so erzeugten Pseudonyme bezüglich Gleichheit. Daher werden deterministische symmetrische Verfahren probabilisiert. Dies geschieht, indem zusätzlich zum symmetrischen geheimen Schlüssel ein frischer Salt-Wert für jedes Klartextdatum gewählt wird. Der Salt wird als Initialisierungsvektor eines sicheren Blockchiffre-Betriebsmodus[2] genutzt. Der Salt wird vorgehalten, muss aber nicht geheimgehalten werden.
Bei der Verwendung von asymmetrischen Verschlüsselungsverfahren müssen deterministische Verfahren aus zwei Gründen probabilisiert werden. Der erste Grund bezieht sich auf die Informationssicherheit. Hintergrund ist, dass auf deterministisch erzeugten Chiffraten Chosen-Plaintext-

[2] siehe z.B. [118] Seite 124

Angriffe möglich sind [16]. Ein Beispiel ein auf diese Weise angreifbares Verschlüsselungsverfahren ist das Rivest-Shamir-Adleman-Verfahren (RSA) ohne Padding. Der zweite Grund bezieht sich auf die mit dem Determinismus einhergehende zusätzliche Nutzbarkeit der Verkettbarkeit bezüglich Gleichheit. In dieser Arbeit sollen Pseudonyme nach Möglichkeit so erstellt werden, dass zusätzlich zur intendierten Nutzbarkeit ein weiterer Informationsabfluss möglichst gering gehalten wird. So soll das Risiko gemindert werden, dass ein Angreifer durch Kenntnisnahme einzelner Pseudonyme mehrere Nutzbarkeiten in Korrelationsangriffen zur Deduktion des Klartextdatums ausnutzt. Neben der Aufdeckbarkeit unter Nutzung des geheimen Schlüssels sind mit deterministischen Verschlüsselungsverfahren erzeugte Pseudonyme ohne Nutzung eines Schlüssels verkettbar bezüglich Gleichheit. Daher müssen die Verschlüsselungsverfahren probabilisiert werden. Dies wird durch die Nutzung von Salts oder Padding umgesetzt. Aus der Anwendung des deterministischen symmetrischen Verschlüsselungsverfahrens in einem sicheren Betriebsmodus[3] mit einem zufälligen Salt als Initialisierungsvektor ergeben sich eindeutige Pseudonyme. Kommt dasselbe Klartextdatum zweimal in der Menge der Eingabedaten vor, so resultieren aus der Anwendung dieses Verfahrens zwei verschiedene Pseudonyme. Somit sind die Pseudonyme probabilistisch erzeugt und durch Entschlüsselung aufdeckbar, jedoch ohne erheblichen Mehraufwand nicht verkettbar bezüglich Gleichheit. Diese Umstände müssen bei dem Entwurf des Pseudonymisierungsverfahrens berücksichtigt werden.

Pseudonymisierungsverfahren

Ausgehend von den vorangegangenen Ausführungen lässt sich das in Algorithmus 1 beschriebene Pseudonymisierungsverfahren herleiten. Für jedes Klartextdatum d aus der zu pseudonymisierenden Klartextdatenmenge wird derselbe symmetrische Schlüssel k zur Verschlüsselung im CBC-Modus[4] verwendet. So kann jedes Subjekt mit Zugriff auf den Schlüssel bei Bedarf die Klartextdaten der zu erzeugenden Pseudonyme aufdecken. Für jedes Klartextdatum wird ein frischer Salt s_d verwendet. Dieser wird als Initialisierungsvektor für den CBC-Modus verwendet. $AES-256$ entspricht dem Stand der Wissenschaft und Technik für deterministische symmetrische Verschlüsselungsverfahren und ist praktisch sicher [19, 33]. Durch Anwendung des Salts und dem CBC-Modus wie beschrieben erfolgt eine Probabilisierung. Der Vorteil gegenüber gängigen asymmetrischen probabilistischen Verfahren ist praktischer Natur. So ist $AES-256$ sehr schnell in der Erzeugung der Pseudonyme. Die Länge der AES-Chiffrate beträgt 128 Bit pro Eingabe-Blockgröße von 128 Bit. Sie ist damit deutlich geringer als die Länge von Chiffraten von asymmetrischen probabilistischen Verfahren mit vergleichbarer Sicherheit. Ein Beispiel hierfür ist Elgamal mit einer verdoppelten Klartextlänge [54].
Ergebnis des Probabilisierungsverfahrens ist eine Menge von Pseudonymen, die zwar aufdeckbar, aber nicht verkettbar bezüglich Gleichheit ist. Das Verfahren ist in Algorithmus 1 gelistet.

[3] Siehe z.B. Kapitel 5 in [118].
[4] Cipherblock-Chaining-Mode (CBC) ist ein Blockmodus für Blockchiffren. Für Details siehe z.B. [118].

Eingabe : Klartextdaten $\{d_1,\ldots,d_n\}$, Schlüssel k.
Ausgabe : Aufdeckbare Pseudonyme mit Salts $\{(p(d_1),s_1),\ldots,(p(d_n),s_n)\}$.

```
1 Menge M = {};
2 for d in {d_1,...,d_n} do
3 |   Erstelle frischen Salt s_d;
4 |   p(d) = CBC_{s_d}(AES-256_k(d));
5 |   M.insert((p(d), s_d));
6 end
7 return M;
```

Algorithmus 1 : Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung Aufdeckbarkeit.

Eingabe : **Analyst:** Aufdeckbares Pseudonym mit Salt $(p(d),s)$,
Dekryptor: Schlüssel k.
Ausgabe : Klartextdatum d.

1 **Dekryptor** berechnet die Entschlüsselung $d := CBC_s^{-1}(AES-256_k^{-1}(p(d)))$ und sendet d an den Analyst;

Algorithmus 2 : Auswertung der Nutzbarkeitsanforderung Aufdeckbarkeit.

Auswertung der Nutzbarkeit

Um nach Algorithmus 1 erzeugte Pseudonyme aufdecken zu können, wird das im Algorithmus 2 gelistete Verfahren angewendet. Hierbei sendet der Analyst das aufzudeckende Pseudonym an den Dekryptor. Der Dekryptor entschlüsselt das Pseudonym unter Anwendung des Entschlüsselungsalgorithmus des AES-256 und dem geheimen Schlüssel k. Der Salt wird hierbei als Initialisierungsvektor bei der Entschlüsselung im CBC-Modus verwendet. Das Ergebnis entspricht dem aufgedeckten Klartextdatum. Dieses wird an den Analysten zurückgegeben.

5.2 Anforderungsklasse Verkettbarkeit bzgl. Relation

In den folgenden drei Unterabschnitten werden exemplarisch Pseudonymisierungsverfahren für die Anforderungsklasse der Verkettbarkeit bezüglich einer Relation beschrieben. Hierfür werden drei Relationen als Ausprägung der Verkettbarkeitsanforderung betrachtet: Die Relation der Gleichheit, die Kleiner-Gleich-Relation und die Elementrelation.

5.2.1 Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Relation Gleichheit

Die Verkettbarkeitsanforderung bezüglich Gleichheit wird in zahlreichen Anwendungen benötigt. So müssen zum Beispiel Datenbankeinträge in Spalten, auf denen eine Join-Operation bei Übereinstimmung der Spalteneinträge ausgeführt werden soll, die Verkettbarkeit bezüglich Gleichheit ermöglichen.
Um bezüglich der Relation Gleichheit verkettbare Pseudonyme erzeugen zu können, können deterministische Verfahren auf den Klartextdaten angewendet werden. Gleiche Klartextdaten werden hierbei auf gleiche Pseudonyme abgebildet. Die Pseudonyme können dann ohne Hinzunahme weiterer Information bezüglich Gleichheit verkettet werden. Um Pseudonyme zu erzeugen, die zwar verkettbar bezüglich Gleichheit, jedoch nicht aufdeckbar sind, werden Hash-Funktionen verwendet. Typischerweise bilden Hash-Funktionen eine vergleichsweise große Eingabemenge auf eine deutlich kleinere Ausgabemenge ab. Dieser Umstand führt zu Kollisionen, d.h. unterschiedliche Eingaben können auf dieselbe Ausgabe abgebildet werden. Daher werden kryptographische Hash-Funktionen [118, 53] verwendet. Diese weisen in der Praxis eine vernachlässigbar geringe Kollisionswahrscheinlichkeit auf. So erzeugte Pseudonyme haben somit in der Praxis die Eigenschaft, dass sie dann und nur dann denselben Wert aufweisen, wenn die zugrundeliegenden Klartextdaten den gleichen Wert haben. Dies gilt für alle Pseudonyme, die mit einer festgelegten Hash-Funktion einer festgelegten Parametrisierung erzeugt wurden. Ein Nachteil dieser Eigenschaft ist, dass bei bekannter Parametrisierung so erzeugte Pseudonyme über beliebige Klartextdaten bezüglich der Gleichheit verkettbar sind. Dies erleichtert gerade bei kleinen Klartexträumen die Inferierung der Klartextdaten aus den Pseudonymen. Ein Beispiel ist die Datenpseudonymisierung von IPv4-Adressen mittels einer Hash-Funktion. IPv4-Adressen haben eine Länge von 32 Bit [124]. Somit beträgt die Größe des Suchraums der Adressen im Klartext 2^{32}. Um ein Klartextdatum aus durch deterministisches Hashing mit bekannter Parametrisierung pseudonymisierten Adressen zu ermitteln, müssen also höchstens 2^{32} mögliche Adressen mit diesem Verfahren gehasht werden und mit der Menge der Pseudonyme auf Übereinstimmung getestet werden. Die Vorgehensweise ist in Algorithmus 3 beschrieben. Wenn die Möglichkeit besteht, die Kandidaten für die Klartextdaten auf eine hinreichend kleine Menge einzugrenzen, kann ein solcher sogenannter Wörterbuchangriff auf durch deterministisches Hashing mit bekannter Parametrisierung pseudonymisierte Adressen effizient durchgeführt werden. Ein Beispiel ist ein Wörterbuchangriff auf mit SHA-256 pseudonymisierte IPv4-Adressen. Verwendet wird ein herkömmlicher Rechner mit einem Intel-Prozessor mit acht Kernen. Ohne Kenntnis des Adress-Präfix beträgt hier die Laufzeit des gesamten Angriffs lediglich etwas mehr als zwei Minuten [82]. Somit bietet das Pseudonymisieren mit deterministischen Hash-Funktionen ohne weitere Parametrisierung insbesondere bei kleinen Suchräumen keinen ausreichenden Vertraulichkeitsschutz.

Pseudonymisierungsverfahren

Um die Verkettbarkeit bezüglich Gleichheit auf eine Pseudonymisierung einer festgelegten Datensammlung einzuschränken, wird die Hash-Funktion parametrisiert. Hierzu wird für jede Menge von Klartextdaten, deren Pseudonymisierung diese Nutzbarkeitsanforderung erfüllen soll, das

Eingabe : n Pseudonyme $p_1,\ldots,p_n$ mit $p_i = Hash(d_i')$, $i \in \{1,\ldots,n\}$ und $n \leq 2^{32}$.
2^{32} Klartextdaten $d_1,\ldots,d_{2^{32}}$.
Ausgabe : Liste $l = [(a,b) \mid a = Hash(b)]$.

```
1  l = [];
2  P = {p_1,...,p_n};
3  for d ∈ {d_1,...,d_{2^32}} do
4  |   a = Hash(d);
5  |   P' := P;
6  |   for p ∈ P' do
7  |   |   if a == p then
8  |   |   |   P' = P' \ {p};
9  |   |   |   l.append((p,d));
10 |   |   end
11 |   end
12 end
13 return l;
```

Algorithmus 3 : Angriff auf pseudonymisierte IPv4-Adressen bei der Verwendung von Hash-Funktion ohne Salt als Pseudonymisierungsverfahren.

Eingabe : Klartextdaten $d_1,\ldots,d_n$, Salt s.
Ausgabe : Gleichheit-verkettbare Pseudonyme $p(d_1),\ldots,p(d_n)$.

```
1  M = {};
2  for d in d_1,...,d_n do
3  |   p(d) = SHA-3(d ⊕ s);
4  |   M.append(p(d));
5  end
6  return M
```

Algorithmus 4 : Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Relation Gleichheit.

folgende Pseudonymisierungsverfahren angewendet: Es wird ein frischer Salt gewählt. Dieser wird mit dem Klartextdatum bitweise Exklusiv-Oder-verknüpft. Das Ergebnis der Exklusiv-Oder-Verknüpfung wird gehasht. Im Ergebnis erhält man Pseudonyme, die zwar untereinander verkettbar bezüglich Gleichheit sind, jedoch nicht mit Klartextdaten, die nicht in der eingangs festgelegten Menge vorhanden sind. Algorithmus 4 listet das Pseudonymisierungsverfahren. Für die Operation `XOR` wird im Algorithmus der Operator $\oplus$ verwendet.

Auswertung der Nutzbarkeit

Für die Auswertung der Nutzbarkeit werden die Pseudonym-Pseudonym-Verkettbarkeit in Algorithmus 5 und die Pseudonym-Klartext-Verkettbarkeit in Algorithmus 6 beschrieben. Für die Pseudonym-Pseudonym-Verkettbarkeit werden die auszuwertenden Pseudonyme auf Gleichheit

Eingabe : Bezüglich Gleichheit zu verkettende Pseudonyme $p_1, \ldots, p_n$ mit $n \geq 2$.
Ausgabe : `true`, falls Klartextdaten gleich, `false` sonst.
1 **return** $p_1 == \ldots == p_n$;

Algorithmus 5 : Auswertung der Nutzbarkeitsanforderung Pseudonym-Pseudonym-Verkettbarkeit bezüglich der Relation Gleichheit.

getestet. Das Ergebnis des Tests kann unmittelbar auf die zugrundeliegenden Klartextdaten übertragen werden.

Für die Pseudonym-Klartext-Verkettbarkeit soll für ein Klartextdatum und eine Menge vorliegender Pseudonyme bestimmt werden, ob das Klartextdatum gleich dem zugrundeliegenden Datum eines Pseudonyms aus der Menge ist. Hierfür wird zunächst das Klartextdatum mit demselben Salt und derselben Hash-Funktion pseudonymisiert, die auch bei der Erstellung der Pseudonyme nach Algorithmus 4 zum Einsatz gekommen sind. Zur Auswertung der Relation wird das Ergebnis analog zur Pseudonym-Pseudonym-Verkettbarkeit auf Gleichheit mit den gegebenen Pseudonymen getestet. Aus jedem Gleichheitstest wird auf die Relation der zugrundeliegenden Klartextdaten geschlossen.

Ähnliche Verfahren wurden bereits in der Literatur beschrieben. So kommt ein solches Verfahren beispielsweise in der Arbeit von Flegel et al. [62] vor. Im Rahmen des Entwurfs von Frühwarnsystemen für Netzwerkangriffe werden bezüglich der Gleichheit verkettbare Pseudonyme zur Verkettung suspekter IP-Adressen erzeugt. Im Unterschied zu der vorliegenden Arbeit basiert das dort beschriebene Pseudonymisierungsverfahren auf der Anwendung einer Hash-Funktion ohne Salt. Daher kann die dort gegebene Verkettbarkeit nicht auf die gegebene Datensammlung eingeschränkt werden. Im Kontext der Analyse von System-Ereignissen werden in [159] ebenfalls bezüglich der Gleichheit verkettbare Pseudonyme beschrieben. Die dort beschriebenen Pseudonymisierungsverfahren basieren auf Message-Authentication-Codes (MAC), die aus kryptographischen Hash-Funktionen mit Schlüssel konstruiert werden[5]. Damit wird die Verfügbarkeit der Nutzbarkeit durch Einsetzen unterschiedlicher Schlüssel gesteuert. In [73] wird analog zu [159] ein ähnliches Pseudonymisierungsverfahren im Kontext des Identitätsmanagements beschrieben.
Es kann festgehalten werden, dass für die Verkettbarkeit bezüglich der Gleichheitsrelation in der Literatur wiederkehrend einander stark ähnelnde Pseudonymisierungsverfahren beschrieben werden. In der vorliegenden Arbeit soll ein Beitrag zur erleichterten Anwendung von Pseudonymisierungsverfahren nach dem Stand der Wissenschaft geleistet werden. Dies soll insbesondere durch die Formulierung von Anforderungsklassen in Kapitel 3 und diesen zugeordneten Pseudonymisierungsverfahren im vorliegenden Kapitel erreicht werden.

[5] Siehe z.B. Kapitel 12 in [118]

Eingabe : Bezüglich Gleichheit verkettbare Pseudonyme $(p_1 := SHA-3(d_1 \oplus s)),\ldots,(p_k := SHA-3(d_k \oplus s))$, Salt s, Klartextdatum d.
Ausgabe : Menge der Pseudonyme, deren Klartextdatum gleich d ist.

```
1 M = {};
2 p(d) = SHA-3(d ⊕ s);
3 for p in p_1,...,p_k do
4 |   if p == p(d) then
5 |   |   M.append(p(d));
6 |   end
7 end
8 return M;
```

Algorithmus 6 : Auswertung der Nutzbarkeitsanforderung Pseudonym-Klartext-Verkettbarkeit bezüglich der Relation Gleichheit.

Eingabe : Klartextdaten $d_1,\ldots,d_n$, geheimer Schlüssel k.
Ausgabe : Ordnungsoffenbarende Pseudonyme $M = \{p_1,\ldots,p_n\}$.

```
1 M = {};
2 for i ∈ {1,...,n} do
3 |   Berechne die binäre Repräsentation b_1 ... b_m von d_i;
4 |   for j ∈ {1,...,m} do
5 |   |   Berechne q_j := F(k,(j,b_1 ... b_{j-1} 0^{m-j})) + b_i  mod 3;
6 |   end
7 |   M.insert((q_1,q_2,...,q_m));
8 end
9 return M;
```

Algorithmus 7 : Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Kleiner-Gleich-Relation.

5.2.2 Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Relation Kleiner-Gleich

Damit Pseudonyme bezüglich der Relation Kleiner-Gleich verkettbar sind, muss aus den Pseudonymen abgeleitet werden können, welche Auswertung der Relation sich auf den zugrundeliegenden Klartextdaten ergibt. Die Relation Kleiner-Gleich ist transitiv. Daher können Daten, auf denen diese Relation bestimmt werden kann, alle miteinander in Relation gesetzt werden. Aus der Erhaltung der Relation Kleiner-Gleich auf den Pseudonymen von Klartextdaten folgt daher, dass aus den Pseudonymen die Ordnung der zugrundeliegenden Klartextdaten ermittelt werden kann. Für das anzuwendende Pseudonymisierungsverfahren bieten sich sogenannte ordnungsoffenbarende Verschlüsselungsverfahren[6] an. Im Vergleich zu den ordnungserhaltenden Verschlüsselungsverfahren[7]

[6] Aus dem Englischen order-revealing encryption; siehe z.B. [39].
[7] Aus dem Englischen order-preserving encryption.

wird die Information, die aus dem Klartextdatum im Chiffrat erhalten bleibt, weiter reduziert. Somit bieten sie einen höheren Vertraulichkeitsschutz.

Pseudonymisierungsverfahren

Das im Folgenden beschriebene Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Relation Kleiner-Gleich wird auf der Basis des Verfahrens von Chenette et al. [39] konstruiert. Im Vergleich zu anderen praktikablen Arbeiten kann der durch die Erhaltung der Nutzbarkeit inhärente Informationsabfluss (engl. *leakage*) besser quantifiziert und eingeschränkt werden und damit der Schutz der Vertraulichkeit erhöht werden [39].
Diese Arbeit wurde in der vergleichenden Arbeit von Bogatov et al. [24] als praktisch sehr effizient eingestuft. Für die Erstellung eines Pseudonyms eines Klartextdatums der Länge m Bit ist eine Verschlüsselung mit dem Aufwand des Aufrufs von m Pseudozufallszahlengeneratoren erforderlich. Die Auswertung der Verkettbarkeit bezüglich der Relation Kleiner-Gleich erfordert im Vergleich zur Ausführung auf den Klartextdaten jedoch keinen zusätzlichen Berechnungsaufwand. Die Größe des Pseudonyms verdoppelt sich im Vergleich zu der des Klartextdatums.
Um ein Klartextdatum d zu pseudonymisieren, wird seine Bit-Repräsentation zunächst in m Werte aufgeteilt. Die Werte werden wie folgt berechnet. Ein jedes Bit b von d wird mit den zu b signifikanteren Bits von d konkateniert. Das Ergebnis wird einer Pseudozufallsfunktion f mit Schlüssel übergeben. Die Ausgabe von f wird dann mit dem zu b nächsten Bit geringerer Signifikanz addiert. Dieses Verfahren wird für jedes der m Bits von d einmal durchgeführt. Die so berechneten Werte ergeben gemeinsam das Pseudonym. Dieses Verfahren ist in Algorithmus 7 gelistet.

Auswertung der Nutzbarkeit

Um gemäß Algorithmus 7 generierte Pseudonyme $p_1 = p(d_1) := \left[q_{11}, q_{12}, \ldots, q_{1k_1}\right]$ und $p_2 = p(d_2) := \left[q_{21}, q_{22}, \ldots, q_{2k_2}\right]$ bezüglich der Relation Kleiner-Gleich vergleichen zu können, werden die Bitwerte q_{1i} und q_{2i} solange paarweise miteinander verglichen, bis eines der beiden Werte um genau 1 größer ist. Dieser Wert gibt den Wert und die Position des ersten Bits an, in dem d_1 und d_2 sich unterscheiden. Daraus kann abgeleitet werden, ob $p_1 < p_2$ gilt oder $p_2 < p_1$. Falls kein Bit-Paar mit einem Werteunterschied von 1 existiert, so sind die zugrundeliegenden Klartextdaten von p_1 und p_2 gleich. Das Verfahren zur Auswertung der Nutzbarkeit als Pseudonym-Pseudonym-Verkettbarkeit ist als Algorithmus 8 gelistet.
Mit dem beschriebenen Pseudonymisierungsverfahren werden Pseudonyme bereitgestellt, die die Nutzbarkeit der Verkettbarkeit bezüglich der Relation Kleiner-Gleich bereitstellen. Nebeneffekt dieser Nutzbarkeit ist, dass die Pseudonyme ebenfalls verkettbar bezüglich Gleichheit sind. Ein weiterer Nebeneffekt ist, dass die Pseudonyme durch die Kodierung der Ordnung die approximative Abschätzung der Distanz zwischen Pseudonymen ermöglichen. Für Einzelheiten zur Korrektheit der Berechnung und dem Sicherheitsniveau wird auf die Arbeit von Chenette et al. [39] verwiesen.

```
Eingabe  : Ordnungsoffenbarende Pseudonyme p_1 := [q_1,...,q_m]
           und p_2 := [r_1,...,r_m].
Ausgabe  : 1, falls p_1 ≤ p_2, 0 sonst.
1  for i ∈ {1,...,m} do
2  |  if q_i == r_i + 1  mod 3 then
3  |  |  return 0;
4  |  end
5  |  else
6  |  |  if q_i + 1  mod 3 == r_i then
7  |  |  |  return 1;
8  |  |  end
9  |  end
10 end
11 return 0;
```

Algorithmus 8 : Auswertung der Nutzbarkeit Verkettbarkeit bezüglich der Kleiner-Gleich-Relation.

5.2.3 Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Elementrelation

Um zu bestimmen, ob ein gegebenes Klartextdatum in einer bei einem Dritten vorliegenden Datenmenge repräsentiert ist, wird die Datenmenge typischerweise nach diesem Klartextdatum durchsucht. Hierfür erhalten Dritte Kenntnis des Klartextdatums. Häufig dürfen personenbezogene Daten nur pseudonymisiert und nicht im Klartext an Dritte weitergegeben werden. Soll dann ermittelt werden, ob eine Menge von gegebenen personenbezogenen Daten in einer Datenmenge vorhanden ist, so ist ein Pseudonymisierungsverfahren erforderlich, welches für ein gegebenes Pseudonym die Frage beantwortet, ob der zugrundeliegende Klartext in der Datenmenge vorhanden ist ohne den Klartext freizugeben. Dies wird als eine Variante des *Private-Set-Intersection* [64, 6] betrachtet. Eine der beiden Mengen enthält hierbei lediglich ein Element, nämlich den Klartext. Das Verfahren kann auch als Variante des *Private-Information-Retrieval* [40] betrachtet werden. Hierbei kann eine Anfrage an eine Datenbank gestellt werden, deren Antwort erfolgt, ohne dass der Datenbank die Antwort bekannt wird. Analog hierzu erfährt der Inhaber der angefragten Menge das angefragte Element nicht.
Im Folgenden wird diese Variante zur Bestimmung der Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Relation „Element von“ beschrieben. Für das Pseudonymisierungsverfahren wird ein zuvor zwischen dem Analyst, dem Pseudonymisierer und dem Dekryptor als einer weiteren, die Pseudonymisierung der Datensammlung vorhaltenden Partei ausgehandelter Schlüssel verwendet. Dieser Schlüssel wird auch für die Auswertung der Nutzbarkeit wiederverwendet.
Das in diesem Abschnitt beschriebene Verfahren basiert auf einer von der Autorin bereits veröffentlichten Arbeit [91]. Der dort verwendete Schlüssel wird innerhalb eines Diffie-Hellman-Schlüsselaustausches zur Etablierung eines gemeinsamen Schlüssels für eine kooperierende Gruppe

Eingabe : Klartextdatum d, privater Schlüssel k, Bloom-Filter-Parameter fp, mf.
Ausgabe : Kodierung eines Pseudonyms von d für den Bloom-Filter $\mathcal{BF}$.
1 Generiere leeren Bloom-Filter $\mathcal{BF}$ mit den Parametern fp, mf;
2 $p := SHA3(d \mid k)$;
3 $c_p = \mathcal{BF}.Insert(p)$;
4 **return** c_p;

Algorithmus 9 : Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Elementrelation

von Anfragenden eines Dienstes verwendet. Das in [91] erarbeitete Szenario wird in Kapitel 7 als Anwendungsbeispiel beschrieben.

Pseudonymisierungsverfahren

Das Pseudonymisierungsverfahren basiert ähnlich zur Pseudonym-Klartext-Verkettbarkeit bezüglich der Relation Gleichheit auf kryptographischen Hash-Funktionen. Ähnlich ist hier auch, dass der Analyst ein Klartextdatum hasht, um ihn mit einer Menge von Pseudonymen zu vergleichen. Der Unterschied ist, dass der Analyst hier keinen Zugriff auf die gesamte Pseudonymisierung erhält. Die Menge der Pseudonyme liegt als Bloom-Filter[23] beim Dekryptor. Diesem sind die einzelnen Pseudonyme nicht bekannt. Ihm liegt lediglich der Bloom-Filter mit den einkodierten Pseudonymen vor. Zum Zeitpunkt der Auswertung lernt der Dekryptor die Kodierungen der Pseudonyme der angefragten Daten. Lediglich der Pseudonymisierer kennt die Klartextdaten. Dieser muss nicht identisch mit dem Dekryptor sein. Weiterhin soll das Klartextdatum, das der Analyst abfragt, dem Dekryptor unbekannt bleiben. Dies stellt eine besondere Herausforderung dar, wenn die Menge der möglichen Klartextdaten klein ist. Der Suchraum ist dann leicht in ein Wörterbuch zu kodieren und entsprechend abzufragen. Dieser Umstand muss bei der Entscheidung für die Umsetzung dieser Nutzbarkeitsanforderung berücksichtigt werden.
Für die Datenpseudonymisierung eines abzufragenden Klartextdatums wird dieses zunächst mit einem zuvor mit dem Pseudonymisierer ausgehandelten Schlüssel konkateniert. Das Ergebnis der Konkatenation wird mit SHA-3 [52] gehasht. Dann wird das Pseudonym in einen leeren Bloom-Filter eingefügt. Der leere Bloom-Filter ist bezüglich der Falsch-Positiv-Rate fp und der maximalen Anzahl der einfügbaren Elemente mf identisch mit dem Bloom-Filter des Dekryptors im leeren Zustand. Daraus ergibt sich die Kodierung des Pseudonyms im Bloom-Filter. Der Analyst fragt diese Kodierung beim Dekryptor an. Dieser gleicht den Bloom-Filter auf ein Vorkommen der Kodierung ab. Dieses Pseudonymisierungsverfahren ist in Algorithmus 9 skizziert.

Auswertung der Nutzbarkeit

Um die Nutzbarkeit „Element von“ auswerten zu können, muss dem Dekryptor die Pseudonymisierung der zu testenden Menge vorliegen. Der Analyst fragt dann bei dem Dekryptor an, ob ein Element in der ihm vorliegenden Menge vorkommt. Der Pseudonymisierer muss nicht identisch

Eingabe : **Pseudonymisierer:** $D = \{d_1, \ldots, d_n\}$, privater Schlüssel k, Parameter für $\mathcal{BF}(P(D))$.
Analyst: c_p.
Ausgabe : 1, falls $c_p \in \mathcal{BF}(P(D))$, 0 sonst.

1 **Vorbereitung Pseudonymisierer:**
2 $\mathcal{BF}(P(D)) = \{\}$;
3 **for** $d \in D$ **do**
4 $\quad p \coloneqq SHA3(d \mid k)$;
5 $\quad \mathcal{BF}(P(D)).insert(p)$;
6 **end**
7 **Pseudonymisierer** sendet $\mathcal{BF}(P(D))$ an Dekryptor;
8 **Abfrage: Analyst** sendet c_p an Dekryptor;
9 **Dekryptor** ermittelt c_p in $\mathcal{BF}(P(D))$;
Result : Dekryptor gibt 1 an Analyst zurück, falls $c_p \in \mathcal{BF}(P(D))$; sonst 0.

Algorithmus 10 : Auswertung der Nutzbarkeit Verkettbarkeit bezüglich der Elementrelation.

mit dem Dekryptor sein. Wenn der Dekryptor nicht identisch mit dem Pseudonymisierer ist, so erstellt letzterer die Pseudonymisierung und sendet diese an den Dekryptor. Der Pseudonymisierer überführt hierzu die Klartextdaten der Menge in eine Bloom-Filter-Repräsentation. Hierfür nutzt er denselben Schlüssel, den auch der Analyst für die Datenpseudonymisierung des abzufragenden Klartextdatums verwendet. Details des Schlüssel- und Bloom-Filter-Parameteraustauschs werden in Kapitel 7.2 beschrieben. Die Verfahren aus [91] werden im Anhang gelistet.
Der Analyst pseudonymisiert das Klartextdatum, das auf Verkettbarkeit bezüglich der Elementrelation mit der Menge überprüft werden soll. Hierzu nutzt er das Pseudonymisierungsverfahren aus Algorithmus 9. Mit der Kodierung c_p fragt er den Dekryptor an. Dieser gleicht den Bloom-Filter der pseudonymisierten Menge auf ein Vorliegen der Kodierung ab. Findet er die Kodierung und somit das Pseudonym in der Menge wieder, gibt er 1 zurück. Der Analyst folgert dann, dass das dem Pseudonym zugrundeliegende Klartextdatum in der Menge vorhanden ist. Ansonsten gibt er 0 zurück. Hieraus folgert der Analyst, dass das Klartextdatum nicht Element der Menge ist. Dieses Verfahren ist in Algorithmus 10 skizziert. Für Einblicke in Laufzeiten und Speicherplatzbedarf der Auswertung der Nutzbarkeit wird auf Abbildungen 3 und 2 des Anhangs verwiesen.

5.3 Anforderungsklasse Operation

In der Datenanalyse besteht die Anforderung, grundlegende Operationen auf den Datenattributen durchführen zu können. Ein Beispiel ist die Berechnung von Durchschnittswerten von Zeitstempeln in Logdaten von Telefonaten, etwa zur Ermittlung von typischen Zeiträumen von betrügerischen Anrufen. Dies erfordert für pseudonymisierte Daten die Möglichkeit, Summen über den Zeitstempel-Werten bilden und diese dann durch die Anzahl der Summanden teilen zu können.

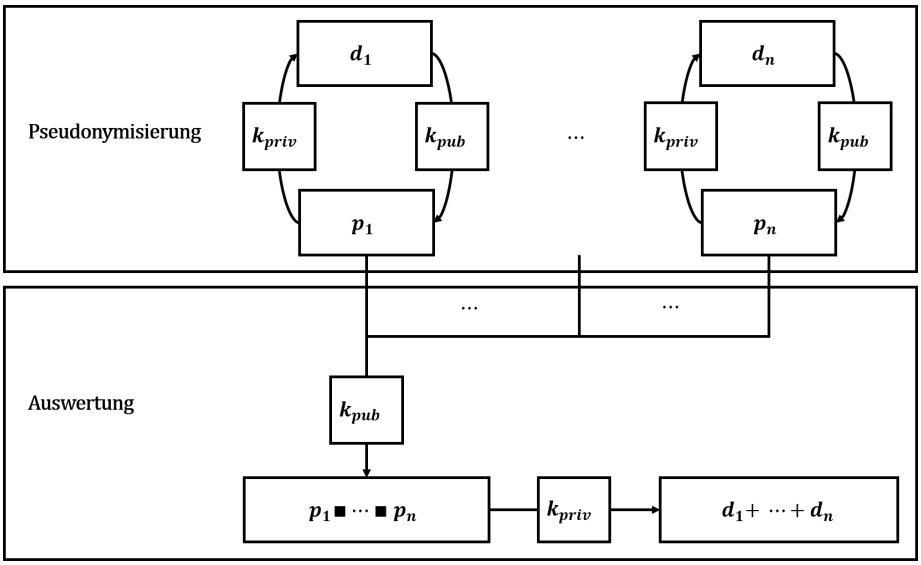

Abbildung 11: *Grundlegender Ansatz der asymmetrischen partiell homomorphen Verschlüsselungsverfahren.*

Um Pseudonyme zu erzeugen, auf denen eine festgelegte Operationen ggf. auch mehrfach durchgeführt werden kann, müssen sich grundlegende Eigenschaften der Klartextdaten in der Pseudonymisierung widerspiegeln. Dies ermöglicht die Ausführung von Verfahren auf den Pseudonymisierungen, die ein zur Durchführung der intendierten Operation auf den Klartextdaten passendes Ergebnis liefern.

Um Pseudonymisierungen zu erzeugen, auf denen den intendierten Operationen entsprechende Berechnungen durchgeführt werden können, wird auf homomorphe Verschlüsselungsverfahren zurückgegriffen. Mit diesen Verfahren erzeugte Chiffrate erlauben die Ausführung bestimmter Operationen auf den Chiffraten. Je nachdem, welche Operationen auf den Chiffraten möglich sind, unterscheidet man zwischen vollhomomorphen, semihomomorphen und partiell homomorphen Verschlüsselungsverfahren [1].
Vollhomomorphe Verschlüsselungsverfahren erlauben die Ausführung von Funktionen $f' : C \longrightarrow C$ auf den Chiffraten C. Ein Beispiel für ein vollhomomorphes Verschlüsselungsverfahren ist das Verfahren von Gentry [65]. Der Raum der Chiffrate ist hierbei auf sogenannten Ideal-Gittern definiert. Analog zur Darstellung von Funktionen in Schaltkreisen ist die Funktion umso komplexer, je mehr Additionen und Multiplikationen zur Darstellung dieser erforderlich sind. Das Ergebnis der Ausführung einer Funktion f' ist ein Chiffrat, welches nach Entschlüsselung das Ergebnis der Ausführung einer zu f' passenden Funktion $f : D \longrightarrow \mathbb{R}$ auf den zugrundeliegenden Klartextdaten freigibt. Ein erheblicher Nachteil aktueller, semantisch sicherer vollhomomorpher Verschlüsselungsverfahren ist die hohe Komplexität und der damit einhergehende große Speicherplatz- und Rechenzeitbedarf. Dies gilt sowohl zum Zeitpunkt der Erstellung der Chiffrate als auch der Durchführung der homomorphen Berechnungen und der Entschlüsselung der Berechnungsergebnisse. Somit sind aktuell bekannte vollhomomorphe Verschlüsselungsverfahren im Allgemeinen nicht praxistauglich [120]. Jedoch werden die Verfahren und deren Implementierung weiter optimiert.

Ein Beispiel ist die *Simple Encrypted Arithmetic Library* (SEAL) [37]. In der Praxis sind sie bereits nutzbar, wenn vergleichsweise kleine Datenmengen verarbeitet werden sollen und die Laufzeit für die Anwendung unkritisch ist. Eine Beispielanwendung sind Demonstratoren für sichere elektronische Wahlsysteme wie z.B. ElectionGuard [156].

Semihomomorphe Verschlüsselungsverfahren[8] basieren auf vollhomomorphen Verfahren. Im Unterschied zu diesen fehlt jedoch der sogenannte Bootstrapping-Schritt. Dieser erwirkt ein Herausrechnen des Rauschens aus den verschlüsselten Berechnungsergebnissen. Dadurch können auf vollhomomorphen Verfahren beliebige Rechenschritte bei ausreichender Rechengenauigkeit durchgeführt werden. Ein Beispiel ist das semihomomorphe Verschlüsselungsverfahren von Brakerski et al. [30]. Mit dem Fehlen des Bootstrappings geht ein im Vergleich zu vollhomomorphen Verfahren reduzierter Ressourcenbedarf einher. Auf den Chiffraten dieser Verfahren können Funktionen ausgeführt werden, die aus einer beliebigen Anzahl von Additionen und einer begrenzten Anzahl Multiplikationen bestehen. Dennoch sind auch die aktuell bekannten semihomomorphe Verfahren im Allgemeinen nicht praxistauglich [26].

Partiell homomorphe Verschlüsselungsverfahren erlauben die Ausführung entweder von beliebig vielen Multiplikationen oder beliebig vielen Additionen. Hier existieren semantisch sichere asymmetrische Verschlüsselungsverfahren. Diese sind im Vergleich zu voll- und semihomomorphen Verfahren bei vergleichbarer Sicherheit in der Praxis effizienter. Für den Entwurf von Pseudonymisierungsverfahren, die die Nutzbarkeit ausgewählter Operationen erhalten, werden daher in dieser Arbeit partiell homomorphe Verschlüsselungsverfahren verwendet.

Um die Nutzbarkeit der für eine Operation $+$ generierten Pseudonyme $p_1, \ldots, p_n$ auszuwerten, wird zunächst auf den Pseudonymen eine für die Operation $+$ geeignete Operation $\blacksquare$ ausgeführt. Hierfür wird der öffentliche Schlüssel k_{pub} verwendet. Das Ergebnis ist ein verschlüsseltes Ergebnis der Ausführung der Operation $+$ auf den Klartextdaten $d_1, \ldots, d_n$. Dieses Berechnungsergebnis wird mit dem privaten Schlüssel k_{priv} entschlüsselt. Die Vorgehensweise ist schematisch in Abbildung 11 dargestellt. Da mit dem privaten Schlüssel auch die Pseudonyme aufgedeckt werden können, darf der Analyst keinen Zugriff auf diesen erhalten. Daher wird hier auf die Rolle des Dekryptors zurückgegriffen. Dieser verwaltet den Zugriff auf den privaten Schlüssel und entschlüsselt vom Analysten angefragte Berechnungsergebnisse. In Abhängigkeit von den Eigenschaften der Operation und dem Pseudonymisierungsverfahren muss sichergestellt werden, dass der Dekryptor keine Pseudonyme aufdeckt. Hierzu ist im Datenverarbeitungssystem die Umsetzung zusätzlicher technisch-organisatorische Schutzmaßnahmen erforderlich, die außerhalb des Fokus des Rahmenwerks dieser Arbeit liegen.

Für die Auswertung der Nutzbarkeiten werden Zwei-Parteien-Protokolle mit den Rollen Analyst und Dekryptor umgesetzt. In dieser Arbeit wird angenommen, dass das Datenverarbeitungssystem und alle beteiligten Rollen die präsentierten Protokolle strikt einhalten und keine weiteren Maßnahmen zur Herausrechnung der Klartextdaten ergreifen. Im Folgenden werden für die beiden exemplarisch umgesetzten Operationen Addition und Multiplikation geeignete Pseudonymisierungsverfahren erarbeitet. Davon ausgehend werden Möglichkeiten zur Auswertung der Operationen als Zwei-Parteien-Protokolle beschrieben.

[8] In der Literatur auch Somewhat- oder Leveled-Homomorphic-Encryption genannt.

5.3.1 Addition

Die Addition als grundlegende Operation ist ein Baustein für eine Vielzahl von Verfahren. Wird sie explizit als Nutzbarkeitsanforderung formuliert, so ist für die Erstellung der Pseudonyme ein geeignetes nutzbarkeitserhaltendes Pseudonymisierungsverfahren erforderlich. In dieser Arbeit wird das Paillier-Verfahren [119] verwendet. Es ist ein additiv-homomorphes, probabilistisches und asymmetrisches Verschlüsselungsverfahren und semantisch sicher gegen gewählte Klartextangriffe (engl. Chosen-Plaintext-Attacks) [119]. Seine Sicherheit basiert auf der Decisional Composite Residuosity-Annahme [119].
Im Folgenden wird das Verfahren skizziert. Gegeben ein Paillier-Schlüsselpaar $(k_{pub}, k_{priv}) = ((N,g),\lambda)$. Zur Erzeugung eines Pseudonyms wird ein einmaliger Zufallswert $r \in (\mathbb{Z}/N\mathbb{Z})$ generiert. Dieser geht in die Berechnung des Pseudonyms ein und wird dann verworfen. Aus Sicherheitsgründen kann r nicht in die Erstellung mehrerer Pseudonyme eingehen. Die Einmaligkeit des in der Berechnung eingesetzten Zufallswerts r hat zur Folge, dass aus zwei gleichen Klartextdaten zwei ungleiche Pseudonyme generiert werden. So kann bei gegebenem öffentlichen Schlüssel und Pseudonym kein Known-Plaintext-Angriff erfolgreich durchgeführt werden.
Zur Addition auf den Klartextdaten sind homomorphe Operationen auf den Chiffraten definiert. Folgerichtig können die Chiffrate auch mit einem konstanten Klartextdatum homomorph multipliziert werden. Dies hat zur Folge, dass die Addition eines Pseudonyms mit einer verschlüsselten 0 bzw. die Multiplikation mit einer 1 im Klartext ein gültiges Pseudonym des ursprünglichen Klartextdatums ergibt. Diese Eigenschaft wird *Malleability* genannt.
Bei der Ausführung von Additionen auf nutzbarkeitserhaltenden Pseudonymen entspricht das Ergebnis der verschlüsselten Summe der den Pseudonymen zugrundeliegenden Klartextdaten. Hierfür müssen zur Pseudonymisierung die Klartextdaten unter Nutzung desselben öffentlichen Schlüssels $k_{pub} := (N,g)$ verschlüsselt werden. Wird nicht für alle Klartextdaten derselbe öffentliche Schlüssel verwendet, so ergibt die Durchführung der homomorphen Addition auf den Pseudonymen nicht die Summe der Klartextdaten.
Um die Summe nutzbar zu machen, muss sie zunächst unter Nutzung des privaten Schlüssels entschlüsselt werden. Hierbei sind besondere Vorsichtsmaßnahmen zu ergreifen. Es gilt zu verhindern, dass die Homomorphie-Eigenschaften und die Eigenschaft der Malleability mittels mehrfacher Entschlüsselungsanfragen verschlüsselter Berechnungsergebnisse zum Aufstellen und Lösen eines linearen Gleichungssystems führt. Dies hätte eine Inferierung der Klartextdaten durch Herausrechnen zur Folge.
Zu den Maßnahmen, die vor einer Aufdeckung der Klartextdaten schützen sollen zählt das Monitoring der durchzuführenden Operationen. Hierfür ist die Kenntnis der durchzuführenden Operationen vor dem Zeitpunkt der Durchführung der Datenanalyse erforderlich. Dies ist in Fällen möglich, in denen mit wechselnden Daten wiederkehrend dieselbe Daten-Analyse durchgeführt werden soll.

Pseudonymisierungsverfahren

Zur Erzeugung der Pseudonyme ist der Zugriff auf den öffentlichen Schlüssel (N,g) erforderlich. Aus der natürlichen Zahl N wird die Größe N^2 des Raums der Chiffrate berechnet. g ist der ganzzah-

Eingabe : Klartextdaten $D = \{d_1, \ldots, d_n\}$, öffentlicher Paillier-Schlüssel $k_{pub} = (N, g)$.
Ausgabe : Pseudonyme $P(D) = \{p_1, \ldots, p_n\}$ mit $p_i * p_j = d_i + d_j$ für alle $i, j \in \{1, \ldots, n\}$.

```
1 P(D) = {};
2 for d_i ∈ {d_1, ..., d_n} do
3 |   Wähle zufälliges r ∈ (ℤ/Nℤ)*;
4 |   p_i = g^{d_i} r^N  mod N^2;
5 |   P(D).insert(p_i);
6 end
7 return P(D);
```

Algorithmus 11 : Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung der Operation Addition.

lige Generator, der als Basis mit dem Klartextdatum d_i als Exponent zur Erzeugung der Pseudonyme verwendet wird. Durch Anmultiplizieren eines mit N potenzierten ganzzahligen Zufallswertes r erhält man ein einzigartiges, additiv homomorphes Pseudonym für das Klartextdatum d_i. Alle unter Nutzung eines öffentlichen Schlüssels (N, g) erzeugten Pseudonyme können nun miteinander addiert werden. Jedoch führt eine Addition dieser Pseudonyme mit anderen, mit einem anderen öffentlichen Schlüssel erzeugten Pseudonymen nicht zu sinnvollen Berechnungsergebnissen der Addition. Das Pseudonymisierungsverfahren wird in Algorithmus 11 gelistet.

Auswertung der Nutzbarkeit

Bei der Auswertung der Nutzbarkeit werden zwei Vorgänge unterschieden. Zunächst der Vorgang der Berechnung der Summe. Dieser erfordert lediglich den Zugriff auf die Pseudonyme und den zugehörigen öffentlichen Schlüssel $k_{pub} = (N, g)$. Das Ergebnis der Berechnung ist die verschlüsselte Summe der den Pseudonymen zugrundeliegenden Klartextdaten. Der zweite Vorgang ist die Entschlüsselung der Summe. Hierfür ist der Zugriff auf den privaten Schlüssel $k_{priv} = \lambda$ erforderlich. Da dieser auch zur Entschlüsselung der Klartextdaten genutzt werden kann, sollte er nicht gemeinsam mit den Pseudonymen abgelegt werden. Auch sollte der Zugriff auf den privaten Schlüssel entsprechend überwacht werden. Das Zwei-Parteien-Protokoll für die Addition beinhaltet also die Durchführung der Berechnungen auf den Pseudonymen durch den Analysten und die Entschlüsselung der Berechnungsergebnisse durch den Dekryptor. Das Auswertungsverfahren ist in Algorithmus 12 gelistet.

5.3.2 Multiplikation

Zur Erzeugung von Pseudonymen, die der Nutzbarkeitsanforderung Multiplikation genügen, wird ein Pseudonymisierungsverfahren beschrieben, dass auf einem multiplikativ-homomorphen Verschlüsselungsverfahren basiert. Zum Einsatz kommt das asymmetrische, partiell-homomorphe Elgamal-Verfahren [54]. Auf den mit dem Elgamal-Verfahren erzeugten Pseudonymen kann eine zur Multiplikation passende Operation ausgeführt werden. Das Ergebnis der Ausführung dieser

Eingabe : **Analyst:** öffentlicher Paillier-Schlüssel $k_{pub} = (N, g)$, Pseudonyme $M = \left\{p_i \mid p_i := g^{d_i} r_i^N \mod N^2\right\}$, $M_\sigma \subseteq M$ Teilmenge der Pseudonyme, deren Summe berechnet werden soll.
Dekryptor: privater Paillier-Schlüssel $k_{priv} = \lambda$.
Ausgabe : Summe der Klartextdaten $S := \sum_{p_i \in M_\sigma} d_i$.

1 $S_p = 0$;
2 **for** $p_i \in M_\sigma$ **do**
3 **Analyst** berechnet verschlüsselte Summe S_p auf der Menge der Pseudonyme M_σ:
$S_p := \prod_{i \in |M_\sigma|} p_i = \prod_{p_i \in M_\sigma} g^{d_i} r_i^N \mod N^2$;

Analyst sendet S_p an Dekryptor;
Dekryptor definiert $L(x) := \frac{(x-1)}{N}$;
Dekryptor berechnet entschlüsselte Summe S aus S_p:

$$
\begin{aligned}
S &= Paillier^{-1}(S_p, \lambda) \\
&= \left(\frac{L(S_p^\lambda \mod N^2)}{L(g^\lambda \mod N^2)} \right) \mod N \\
&= \left(\frac{L((\prod_{p_i \in M_\sigma} g^{d_i} r_i^N)^\lambda \mod N^2)}{L(g^\lambda \mod N^2)} \right) \mod N \\
&= \sum_{p_i \in M_\sigma} d_i \mod N.
\end{aligned}
$$

4 **end**
5 **Dekryptor: return** S.

Algorithmus 12 : Auswertung der Nutzbarkeit Addition.

Operation ist ein verschlüsseltes Produkt der den Pseudonymen zugrundeliegenden Klartextdaten. Unter der Computational-Diffie-Hellman-Vermutung [15] und der Annahme, dass der diskrete Logarithmus in gewissen zyklischen Gruppen schwer zu berechnen ist [104], ist das Verfahren semantisch sicher.

Pseudonymisierungsverfahren

Zur Erzeugung der Pseudonyme ist ein Zugriff auf den öffentlichen Schlüssel erforderlich. Dieser besteht aus einem Wert, dessen Basis der Generator der gewählten Gruppe $\mathbb{Z}/q\mathbb{Z}$ des Elgamal-Verfahrens ist. Die Ordnung der Gruppe $\mathbb{Z}/q\mathbb{Z}$ ist die Primzahl q. Der Exponent ist der geheime Schlüssel. Dieser wiederum ist ein zufällig gewählter Wert aus $\mathbb{Z}/q\mathbb{Z}$. Es wird zunächst ein einmaliger Zufallswert $r \in \mathbb{Z}/q\mathbb{Z}$ erzeugt. Dieser wird nun zur Erzeugung der beiden Komponenten des Chiffrat-Tupels genutzt. Die erste Komponente ist der mit r potenzierte Generator. So bleibt r geheim, kann aber gleichzeitig zur Randomisierung in die Verschlüsselung mit einfließen. Die zweite Komponente des Chiffrats wird erzeugt, indem der öffentliche Schlüssel mit r potenziert und dann an das Klartextdatum anmultipliziert wird. Auf diese Weise erhält man ein verschlüsseltes Klartextdatum, der ebenso randomisiert ist. Das Pseudonym eines Klartextdatums ist nun

Eingabe : Klartextdaten $d_1, \ldots, d_n$, öffentlicher Elgamal-Schlüssel $k_{pub} := [g^a]$ mit $k = (k_{pub}, k_{priv})$, $k_{priv} := a \in \mathbb{Z}/q\mathbb{Z}$ zufällig gewählt.
Ausgabe : Pseudonyme $p_1, \ldots, p_n$ mit $p_i \circ p_j = d_i * d_j$.

```
1 M = {};
2 for i ∈ {1,...,n} do
3 |   Wähle zufälliges r ∈ ℤ/qℤ;
4 |   Berechne c_0(d_i) = g^r ∈ G, wobei G eine zyklische Gruppe ist;
5 |   Berechne c_1(d_i) = [g^a]^r · d_i ∈ G;
6 |   p_i := (c_0(d_i), c_1(d_i));
7 |   M.insert(p_i);
8 end
9 return M;
```

Algorithmus 13 : Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung der Operation Multiplikation.

das erzeugte Chiffrat-Tupel. Das Pseudonymisierungsverfahren ist in Algorithmus 13 gelistet. Im Algorithmus sind die Operationen auf den Pseudonymen, die eine Multiplikation der zugrundeliegenden Klartextdaten bewirken, mit dem Operator $\circ$ kenntlich gemacht. Die Multiplikation auf den Klartextdaten ist mit $*$ gekennzeichnet.

Auswertung der Nutzbarkeit

Um die Nutzbarkeit auswerten zu können, benötigt der Analyst den öffentlichen Schlüssel $k_{pub} := [g^a]$. Er nutzt den öffentlichen Schlüssel zur Durchführung der homomorphen Multiplikation. Das verschlüsselte Produkt der den ausgewählten Pseudonymen zugrundeliegenden Klartextdaten entsteht durch komponentenweise Multiplikation. Alle ersten Komponenten der Pseudonyme werden miteinander multipliziert. Das Ergebnis ist die erste Komponente des Chiffrats des verschlüsselten Produkts. Es entspricht dem Generator der mit der Summe aller Zufallszahlen der einzelnen Pseudonyme potenziert wurde. Alle zweiten Komponenten der Pseudonyme werden ebenfalls miteinander multipliziert. Das Ergebnis ist die zweite Komponente des Chiffrats des verschlüsselten Produkts der den Pseudonymen zugrundeliegenden Klartextdaten. Die zweite Komponente beinhaltet die „verfremdeten" Klartextdaten. Der Analyst sendet das verschlüsselte Produkt der Klartextdaten an den Dekryptor. Dieser kann nun unter Zuhilfenahme der ersten Komponente des Produkts und unter Kenntnis des geheimen Schlüssels die zweite Komponente zum Klartext-Produkt entschlüsseln. Die Auswertung der Nutzbarkeit Multiplikation ist in Algorithmus 14 gelistet.

5.4 Anforderungsklasse Algorithmus

In den vorangegangenen Abschnitten dieses Kapitels werden Nutzbarkeitsanforderungen vorgestellt, die einzelne gleichartige Operationen auf den Pseudonymen zur Folge haben. Soll auf

Eingabe : **Analyst:** öffentlicher Schlüssel g^a, Pseudonyme $P := \{p_1, \ldots, p_n\} \subset G \times G$, G ist eine Gruppe der Ordnung q; $p_i := (c_0(d_i), c_1(d_i))$, $c_0(d_i) := g^{r_i}$, $c_1(d_i) := g^{a \cdot r_i} \cdot d_i$; $p_i \circ p_j$ ist eine Operation, die zu $d_i * d_j$ auf den Klartextdaten korrespondiert.
Dekryptor: privater Schlüssel $a \in \mathbb{Z}/q\mathbb{Z}$.
Ausgabe : Für $N := \left\{p_{l_1}, \ldots, p_{l_h}\right\} \subseteq \{p_1, \ldots, p_n\}$: Produkt $\prod_{p_i \in N} d_i$.

1 $prod0 = 1$;
2 $prod1 = 1$;
3 **Analyst** wählt Menge N der Pseudonyme aus P, für deren Klartextdaten das Produkt bestimmt werden soll ;
4 **Analyst** berechnet Produkt der Elemente aus N:
5 **for** $p_i \in N$ **do**
6
$$\begin{aligned} prod0 &= prod0 * c_0(d_i) \\ &= prod0 * g^{r_i}. \\ prod1 &= prod1 * c_1(d_i) \\ &= prod1 * g^{a \cdot r_i} \cdot d_i. \end{aligned}$$
7 **end**
8 **Analyst** sendet das verschlüsselte Produkt $(prod0, prod1)$ an Dekryptor;
9 **Dekryptor** berechnet die Entschlüsselung:
10
$$\begin{aligned} prod &= prod0^{-a} \cdot prod1 \\ &= prod0^{q-a} \cdot prod1. \end{aligned}$$
11 **return** *prod, der Klartext des Produkts der Klartextdaten der Pseudonyme aus* $N \subseteq P$.

Algorithmus 14 : Auswertung der Nutzbarkeit der Operation Multiplikation.

den Pseudonymen ein aus einer Vielzahl unterschiedlicher Operationen bestehender Algorithmus ausgeführt werden können, so kann man den Algorithmus grundsätzlich als eine Menge von Nutzbarkeitsanforderungen auffassen. Diese einzelnen Anforderungen würden dann in einzelnen Pseudonymen umgesetzt werden. Die Gesamtheit der so erstellten nutzbarkeitserhaltenden Pseudonyme würde dann eine für den Algorithmus nutzbarkeitserhaltende Pseudonymisierung ergeben. Mit einem geeignet entworfenen Protokoll kann der Algorithmus auf der Pseudonymisierung ausgeführt werden. Bei der Erstellung einer nutzbarkeitserhaltenden Pseudonymisierung für einen Algorithmus muss jedoch beachtet werden, dass jedes Pseudonym mitunter deutlich mehr Nutzbarkeit aufweist als für die Ausführung des Algorithmus erforderlich ist. Ein Ziel der Datenpseudonymisierung von personenbezogenen Klartextdaten ist jedoch die Datenminimierung. So soll das Risiko gemindert werden, dass die Eigenschaften der Pseudonymisierung zur Inferierung personenbezogener Daten genutzt werden können. Übertragen auf die Nutzbarkeitserhaltung wird daher angestrebt, Pseudonymisierungen so zu generieren, dass sie gerade die Nutzbarkeiten erhalten, die für die Erfüllung der Nutzbarkeitsanforderungen unbedingt erforderlich sind. Die Pseudonymisierungsverfahren sollen daher so entworfen werden, dass möglichst wenig weitere Nutzbarkeiten durch die erzeugten Pseudonymisierungen ermöglicht werden. Um die Nutzbarkeitsanforderung auswerten zu können, werden für die auszuführenden Algorithmen Privatsphäre

erhaltende Varianten entwickelt, die auf den pseudonymisierten Daten ausgeführt werden können. Diese sind meist als Zwei- oder Mehrparteienprotokolle entworfen. Durch die Konstruktion der Protokolle und den Einsatz kryptographischer Verfahren soll neben der Erhaltung der Nutzbarkeit das Risiko einer Reidentifizierung durch Nutzung der Pseudonymisierungsverfahren gemindert werden.

5.4.1 Beispiel k-Means

Für die exemplarische Umsetzung eines auf nutzbarkeitserhaltenden Pseudonymen ausführbaren Algorithmus wird im vorliegenden Abschnitt das k-Means-Clustering-Verfahren [101] beschrieben. Als Verfahren des unüberwachten Lernens kommt es in einer Vielzahl von Anwendungen vor [20]. Im Folgenden wird das Verfahren auf Klartextdaten zusammengefasst. Im ersten Schritt wird die Zahl der gewünschten Cluster k festgelegt. Dann werden entsprechend k initiale Clusterzentren gewählt. In der ursprünglichen Version des Algorithmus sind dies Zufallswerte. In einer jeden Iteration werden die Daten den bezüglich der euklidischen Distanz nächsten Clusterzentren zugeordnet. Im Laufe der Iterationen des Algorithmus wird das Zentrum eines Clusters als Mittelwert der dem Cluster in dieser Iteration zugeordneten Daten gebildet. Das Verfahren terminiert, sobald sich die Werte der Clusterzentren nicht mehr signifikant verändern. Das Verfahren ist in Algorithmus 15 gelistet.

Ein Privatsphäre-schützender k-Means-Algorithmus

Für das k-Means-Clustering-Verfahren ist eine Vielzahl von Privatsphäre schützenden Varianten bekannt [106, 110]. Die Ausführung erfolgt über Mehrparteienprotokolle. Hierfür werden die Daten aus Gründen der Sicherheit mitunter als partitioniert betrachtet. Bei der horizontalen Partitionierung[9] liegt die Datensammlung als Aufteilung von Teilmengen von Datensätzen bei mindestens zwei Parteien. Bei der vertikalen Partitionierung[10] wird die Datensammlung nach Attributen auf mindestens zwei Parteien aufgeteilt. Jede Partei hält dann alle Werte einer Teilmenge der Menge der Attribute der Datensammlung. Für die praktische Umsetzung erfordern diese Verfahren eine stark angepasste Rechenumgebung mit mehreren, miteinander protokollkonform kollaborierenden Parteien ausreichender Rechen- und Kommunikationskapazität. In [110] müssen zum Beispiel verschiedene Parteien ihre Daten pseudonymisiert vorhalten und Teiloperationen des k-Means-Verfahrens ausführen. Sie können daher die Berechnung nicht vollständig an Systeme mit mehr Rechenkapazität auslagern. Ein für die Datenquellen interaktions- und rechenarmes Teilen der pseudonymisierten Daten ist so nicht möglich. Auch setzen die Verfahren eine erhöhte Speicherkapazität bei mehr als einer Partei voraus.
In dem vorliegenden Abschnitt wird ein Verfahren beschrieben, das ohne eine Partitionierung der Daten auskommt. Es wurde innerhalb der Forschungsarbeiten zu dieser Dissertation im Rahmen der Betreuung einer Bachelorarbeit [95] und einer Masterarbeit [96] erarbeitet. Die Daten können nach der Pseudonymisierung dem Analysten übergeben werden. Dieser kann die Berechnungen in Zusammenarbeit mit dem Dekryptor durchführen. Hierfür ist die Rechenlast für den

[9] siehe z.B. [83]
[10] siehe z.B. [48, 110]

Eingabe : Klartextdaten $d_1,\dots,d_n$, Anzahl der Cluster k, Abbruchkriterium ϵ.
Ausgabe : Clustering $C = (C_i)_{i=1,\dots,k}$.

(1) **for** $i = 1,\dots,k$ **do**
  Wähle zufälliges initiales Clusterzentrum $m_i^{(0)}$;
  $C_{m_i} = \{\}$;
end
$l := 1$;
$m_i^{(-1)} := 0$;
while $dist(m_i^{(l)}, m_i^{(l-1)}) \geq \epsilon$ **do**
  (2) **for** d *in* $\{d_1,\dots,d_n\}$ **do**
    Ordne d dem Cluster C_i mit dem nächsten Clusterzentrum m_i zu:
    $C_i = \left\{d \mid dist(d,m_i) \leq dist(d,m_j) \forall j \in \{1,\dots,k\}\right\}$;
  end
  (3) **for** $i \in \{1,\dots,k\}$ **do**
    Bestimme die aktualisierten Clusterzentren:
    $m_i^{(l)} = \frac{\sum_{d \in C_i} d}{|C_i|}$;
  end
  $l++$;
end
return *Clustering* $C = (C_i)_{i=1,\dots,k}$;

Algorithmus 15 : k-Means-Clusteringverfahren auf Klartextdaten nach Kapitel 12 in [121].

Dekryptor im Vergleich zu der des Analysten gering. Dies hat den Vorteil, dass der Dekryptor durch möglicherweise kostenintensive, relativ ressourcenarme Schutzmechanismen wie Trusted-Execution-Environments[11] gehärtet werden kann. Ein weiterer Vorteil gegenüber der Mehrzahl der in der Literatur bekannten Verfahren ist, dass die Klartextdaten bei Bedarf vollständig aufgedeckt werden können. Dies ermöglicht z.B. die Kombinierung der Nutzbarkeitsanforderungen des Algorithmus k-Means und der Aufdeckbarkeit anhand bestimmter Clusterzuordnungen. Weiterhin übertrifft das Verfahren ähnliche Arbeiten hinsichtlich der Laufzeit [96]. Daher ist das Verfahren für den Einsatz im Rahmen der vorliegenden Arbeit geeignet.

Pseudonymisierungsverfahren Ausgangssituation ist eine Datensammlung D, die aus Datensätzen $d_1,\dots,d_m$ besteht. Jeder Datensatz d_i aus D besteht aus numerischen Attributwerten $d_{i,1},\dots,d_{i,n}$. Der k-Means-Algorithmus soll auf einer Pseudonymisierung $P(D)$ von D ausgeführt werden können.
Der Pseudonymisierer erstellt $P(D)$ aus den gegebenen m Klartextdatensätzen d_i. Diese besteht für jeden der Datensätze aus n additiv homomorphen Paillier-Chiffraten der Attributwerte und einer Prüfsumme. Für die Verschlüsselung wird ein Paillier-Schlüsselpaar (k_{pub}, k_{priv}) aus einem öffentlichen k_{pub} und einem geheimen Schlüssel k_{priv} verwendet. Die Prüfsumme eines Datensatzes d_i wird

[11] Siehe z.B. die Software-Guard-Extensions (SGX) von Intel [42].

Eingabe : m Klartextdatensätze $D := (d_1, \ldots, d_m)$ mit $d_i := (d_{i,1}, \ldots, d_{i,n})$,
Gewichte $W := (w_1, \ldots, w_n)$, öffentlicher Paillier-Schlüssel $k_{pub} := (N, g)$.
Ausgabe : Pseudonymisierung $P(D) := \{p_i := p(d_i), \ldots, p_m := p(d_m)\}$ mit
$p_i := (p_{i,1}, \ldots, p_{i,n}, pcs_i)$.

```
1  P(D) := {};
2  for i = 1,...,m do
3  |   sum = 0;
4  |   p_i = [];
5  |   for j = 1,...,n do
6  |   |   Erzeuge die Paillier-Verschlüsselung p_{i,j} von d_{i,j}:
7  |   |   Wähle zufälliges r ∈ (ℤ/Nℤ);
8  |   |   p_{i,j} = g^{d_{i,j}} r^n  mod  N^2;
9  |   |   Füge die Paillier-Verschlüsselung p_i als Komponente hinzu:
10 |   |   p_i.append(p_{i,j});
11 |   |   Ergänze die Prüfsumme sum = sum + d_{i,j} · w_j;
12 |   end
13 |   Verschlüssele die Prüfsumme sum unter Nutzung des öffentlichen
   |    Paillier-Schlüssels k_pub zu pcs_i und füge sie p_i als Komponente hinzu:
14 |   p_i.append(pcs_i);
15 |   Füge p_i der Pseudonymisierung hinzu:
16 |   P(D).insert(p_i);
17 end
18 return P(D);
```

Algorithmus 16 : Pseudonymisierungsverfahren für die Nutzbarkeitsanforderung Algorithmus k-Means-Clustering.

durch das komponentenweise Multiplizieren von n geheimen, großen Zufallszahlen $w_1, \ldots, w_n$ aus $\mathbb{N}$ an die Attributwerte $d_{i,j}$ des Datensatzes gebildet. Mit der Prüfsumme wird überprüft, ob vom Dekryptor zu entschlüsselnde Zwischenergebnisse der Berechnungen korrekt und unverfälscht sind. Diese wird verschlüsselt zu einem zusätzlichen Attribut pcs_i des pseudonymisierten Datensatzes p_i. Diese Prüfsumme wird zum Ausführungszeitpunkt zur Überprüfung der Korrektheit von Zwischenberechnungen verwendet. So soll sichergestellt werden, dass auf den Pseudonyme tatsächlich die vorgesehenen Operationen des k-Means-Verfahrens ausgeführt wurden. In den in der Auswertung folgenden Berechnungen ist in den Algorithmen der Suffix des Bezeichners der Prüfsumme stets cs.
Der Analyst erhält den öffentlichen Schlüssel k_{pub} und $P(D)$. Der Dekryptor erhält den privaten Schlüssel k_{priv}, und die Gewichte $w_1, \ldots, w_n$. Das Pseudonymisierungsverfahren ist in Algorithmus 16 beschrieben.

Auswertung der Nutzbarkeit Die Auswertung der Nutzbarkeit ist die Ausführung des Privatsphäre schützenden k-Means als Zwei-Parteien-Protokoll. Es wird angenommen, dass der Pseudonymisierer mit dem Schlüsselmanager identisch ist. Der Dekryptor erhält die geheimen Parameter k_{priv} und $w_1, \ldots, w_n$. Seine Rolle ist es, innerhalb von Analyseschritten Berechnungen

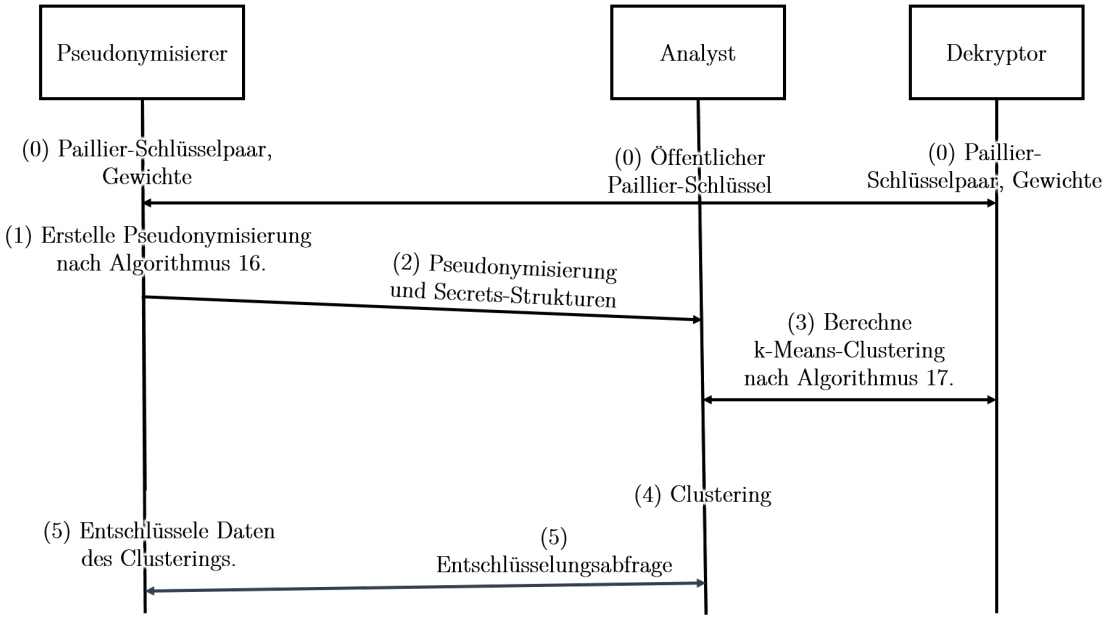

Abbildung 12: *Überblick des Ablaufs der Auswertung der Nutzbarkeit des Algorithmus k-Means auf Chiffraten.*

durchzuführen, die einen Zugriff auf geheime Parameter erfordern. Der Analyst erhält von dem Pseudonymisierer den öffentlichen Schlüssel k_{pub} und die Pseudonymisierung. Immer dann, wenn die Berechnungen einen Zugriff auf geheime Parameter erfordern, interagiert der Analyst entsprechend mit dem Dekyptor. Der Dekryptor kann eine Trusted-Execution-Environment (TEE) wie SGX [42] sein, die auf dem System des Analysten angesiedelt ist. Eine Übersicht dieses Ablaufs ist in Abbildung 12 schematisch dargestellt.
Der Analyst bestimmt im ersten Schritt die initialen Clusterzentren. Hierzu wählt er zufällig k pseudonymisierte Datensätze aus der Pseudonymisierung aus.
Dann ordnet er in Schritt 2 jeden pseudonymisierten Datensatz dem nächsten Clusterzentrum zu. Hierfür müssen Distanzen zwischen den pseudonymisierten Datensätzen und entsprechend auf den verschlüsselten Datenattributen berechnet werden. Die hierfür durchzuführenden Additionen und Subtraktionen entsprechen der in Algorithmus 12 beschriebenen und in [119] eingeführten Addition (und entsprechend Subtraktionen) auf partiell-additiv homomorphen Chiffraten. In den Algorithmen 18 – 22 wird die entsprechende Addition mit dem Operator $\boxplus$ und die Subtraktion mit dem Operator $\boxminus$ gekennzeichnet. Zwischenergebnisse müssen entschlüsselt werden, um Multiplikationen und Divisionen durchführen und minimale Distanzen bestimmen zu können. Hierfür interagiert der Analyst (Algorithmus 19) mit dem Dekryptor (Algorithmus 21).
Letzterer überprüft zunächst mittels der Prüfsummen, ob die Berechnungen den intendierten entsprechen und damit erlaubt sind. Ist dies der Fall, bestimmt der Dekryptor die minimalen Distanzen und gibt diese an den Analysten zurück. Der Analyst ordnet die Distanz über einen für diese Abfrage gewählten Index dem Datum zu. Dann weist der Analyst das Datum dem Clusterzentrum mit der geringsten Distanz zu.

In Schritt 3 aktualisiert der Analyst die Clusterzentren (Algorithmus 20). Wegen der erforderlichen Division unter Nutzung des privaten Schlüssels durch die Anzahl der Cluster interagiert er entsprechend mit dem Dekryptor. Dieser überprüft die Berechnungen anhand der Prüfsummen auf Plausibilität und berechnet die Clusterzentren aus den vom Analyst bereitgestellten Daten (Algorithmus 22).
Es wird solange über Schritt 2 und 3 iteriert, bis die Abbruchbedingung eintritt. Die Berechnung der Iteration erfolgt in Algorithmus 18 gemeinsam durch den Analyst und den Dekryptor. Auch hier werden die Prüfsummen genutzt, um die Protokollkonformität und damit auch die Korrektheit der durchgeführten Berechnungen nachzuvollziehen.
Insgesamt berechnet der Analyst so ein verschlüsseltes Clustering auf den Pseudonymen. Dieses gibt er nach Erfüllung der Abbruchbedingung an den Pseudonymisierer zurück. Dieser kann nun die Pseudonyme entschlüsseln und ggf. Erkenntnisse aus dem Clustering mit dem Analysten teilen.

Anmerkung zur Sicherheit des Auswertungsverfahrens Der Dekryptor ist als Trusted-Execution-Environment (TEE) in der Lage, entsprechend seiner Konfiguration signierten, unveränderlichen Programmcode auszuführen. Daher wird im Folgenden angenommen, dass der Dekryptor protokollkonform agiert.
Der Dekryptor erhält das Paillier-Schlüsselpaar. Dem Analysten wird vertraut, dass er protokollkonform agiert. Der Analyst interagiert mit dem Dekryptor immer dann, wenn Distanzen von Daten zu Clusterzentren zur Bestimmung der Clusterzuordnung entschlüsselt werden müssen. Der Dekryptor lernt dadurch weder die Clusterzentren noch die Klartextdaten. So wird sichergestellt, dass der Analyst keinen Zugriff auf Berechnungsergebnisse im Klartext erhält. Damit wird das Risiko reduziert, dass der Analyst die Berechnungsergebnisse zur Ermittlung der Klartextdaten und somit zu einer unintendierten Aufdeckung durch die Pseudonyme für k-Means nutzt. Dem Dekryptor wird vertraut, dass es das Protokoll einhält. Insbesondere interagiert er nicht über das Protokoll hinaus mit dem Analysten. Dadurch, dass der Dekryptor nur Distanzen ohne Zuordnungen zu den Daten erhält, kann es keine weiteren Erkenntnisse über die Klartextdaten ermitteln. Der dem Dekryptor für die Indexierung der verschlüsselten Distanzen mitgeteilte Wert kann von diesem nicht protokollkonform zugeordnet werden.
Ein weiterer sicherheitsfördernder Mechanismus ist die Prüfsumme. Diese wird aus hinreichend großen, für jedes Datenattribut einzeln zu wählenden Gewichten $w_1, \ldots, w_n$ gebildet. Durch die Zusammensetzung aus Zufallszahlen wird sichergestellt, dass die Prüfsumme nicht rekonstruiert werden kann. Sie wird in den Algorithmen verwendet, um zu überprüfen, ob ein zu entschlüsselnder Wert aus einer im Rahmen des k-Means erlaubten Berechnung stammt. So soll verhindert werden, dass der Dekryptor im Rahmen der Entschlüsselung von Zwischenergebnissen zur unerlaubten Aufdeckung von Klartextdaten missbraucht wird. Dies ist erforderlich, weil durch die Homomorphie-Eigenschaft des Paillier-Verfahrens herkömmliche, auf dem Signieren des Chiffrats basierende Verfahren zur Integritätsprüfung nicht genutzt werden können. Beispiele für alternative Verfahren sind Merkle-Tree- und Blockchain-basierte Verfahren [109]. Mit diesen können die ausgeführten Operationen nachvollziehbar protokolliert werden.

Eingabe : **Analyst:** Nach Algorithmus 16 pseudonymisierte Daten $P(D) = \{p_1, \ldots, p_m\}$, Anzahl der Cluster k, öffentlicher Schlüssel k_{pub}.
Dekryptor: öffentlicher Schlüssel k_{pub}, privater Schlüssel k_{priv}, Gewichte $w_1, \ldots w_n$.
Ausgabe : Clustering C von $P(D)$.

```
(1) Bestimme initiale Clusterzentren durch zufällige Auswahl von k Elementen aus P(D);
while Algorithmus 18 gibt 0 zurück do
    (2) Ordne alle pseudonymisierten Daten p_i aus P(D) dem nächsten Cluster zu
     (Analyst: Algorithmus 19, Dekryptor: Algorithmus 20);
    (3) Passe die Clusterzentren an (Analyst: Algorithmus 21, Dekryptor: Algorithmus
     22);
end
return C;
```

Algorithmus 17 : k-Means auf pseudonymisierten Daten als Zusammensetzung der Algorithmen 18-22.

Die weiteren Verfahren zur Auswertung der Nutzbarkeit Algorithmus k-Means sind im Folgenden aufgeführt. In Algorithmus 17 ist die Zusammensetzung des k-Means aus den einzelnen Verfahren gelistet. Für Einblicke in die Laufzeit der Auswertung der Nutzbarkeit wird auf Abbildungen 3 und 4 des Anhangs verwiesen.

5.5 Fazit zu Pseudonymisierungsverfahren

In dem vorliegenden Kapitel werden Pseudonymisierungsverfahren beschrieben, die in der vorliegenden Arbeit bei der automatisierten Erstellung von nutzbarkeitserhaltenden Pseudonymisierungen zum Einsatz kommen. Ausgehend von den in den vorangegangenen Abschnitten erarbeiteten Erkenntnissen zu den Pseudonymisierungsverfahren werden wesentliche, für die automatisierte Erstellung relevante Eigenschaften zusammengefasst. Diese werden im Folgenden beschrieben.
Für die Erstellung der Pseudonymisierungsverfahren kommen in Abhängigkeit der zu erhaltenden Nutzbarkeit verschiedene Verfahren zum Einsatz. Um die Nutzbarkeit auf der Pseudonymisierung auszuwerten, werden für eine Reihe von Verfahren Mehrparteienprotokolle eingesetzt. Dies ist immer dann der Fall, wenn die Auswertung den Zugriff auf geheime Schlüssel erfordert. Diese Protokolle verhindern den Zugriff einer nicht vertrauenswürdigen Partei auf die geheimen Schlüssel und damit auf ungewollte Nutzbarkeiten. Der Dekryptor aus Kapitel 1.2.1 als eine dritte vertrauenswürdige Partei (TTP, englisch Trusted Third Party) kontrolliert die Freigabe der Nutzbarkeiten auf dem verarbeitenden System. Dies beinhaltet die sichere Verwaltung der geheimen Schlüssel.

Anforderungsklasse Aufdeckbarkeit Für das Pseudonymisierungsverfahren für die Aufdeckbarkeit wird ein probabilistisches symmetrisches Verschlüsselungsverfahren verwendet. Für die

Eingabe : **Analyst:** k nach Algorithmus 16 pseudonymisierte alte Clusterzentren $C^{alt} = \left\{pc_1^{alt},\ldots,pc_k^{alt}\right\}$ mit $pc_i^{alt} := (pc_{i,1}^{alt},\ldots,pc_{i,n}^{alt},pcs_i^{alt})$, k pseudonymisierte aktuelle Clusterzentren $C = \{pc_1,\ldots,pc_k\}$ mit $pc_i := (pc_{i,1},\ldots,pc_{i,n},pcs_i)$.
Dekryptor: Privater Schlüssel k_{priv} und Gewichte (w_j) mit $1 \leq j \leq n$.
Ausgabe : 1, falls Abbruchbedingung erfüllt; 0 sonst.

Analyst:
for $i = 1,\ldots,k$ **do**
  Berechne die Distanz $dist_j$ zwischen pc_j^{alt} und pc_j komponentenweise als $\boxminus$:
  $(pc_{j,1}^{alt} \boxminus pc_{j,1})$;
  if $i == k-1$: **then**
    $dist_{cs} := pcs_i^{alt} \boxminus pcs_i$;
  end
  Sende die k berechneten Distanzvektoren: $dist_j := (dist_1,\ldots,dist_n,dist_{cs})$ an Dekryptor ;
  Dekryptor:
  for $j = 1,\ldots,k$ **do**
    Entschlüssele $dist_j$ zu δ_j;
    Teste die Prüfsumme auf Korrektheit:
    if $\sum_{i=1}^{n} \delta_{j,i} \cdot w_j == \delta_{cs}$ **then**
      Berechne die euklidische Distanz im Klartext:
      $\delta = \sqrt{\sum_{i=1}^{n} \delta_{j,i}^2}$;
    else if $\delta > \epsilon$ **then**
      Sende 0 an Analyst;
    else
      Sende 1 an Analyst;
    end
  end
end

Algorithmus 18 : Auswertung k-Means-Clustering: Überprüfung der Abbruchbedingung in einer k-Means-Iteration.

Auswertung wird eine Aufdeckungsanfrage an den Dekryptor gesendet. Diese gibt die Klartextdaten für erlaubte Aufdeckungsanfragen durch Entschlüsselung frei.

Anforderungsklasse Verkettbarkeit bezüglich einer Relation Für das Pseudonymisierungsverfahren für die Verkettbarkeit bezüglich der Gleichheitsrelation wird eine deterministische Hash-Funktion mit Salt eingesetzt. Für die Auswertung werden die Pseudonyme auf Gleichheit getestet. Daraus wird auf die Gleichheit der Klartextdaten geschlossen.
Für die Verkettbarkeit bezüglich der Relation Kleiner-Gleich wird ein ordnungsoffenbarendes Verschlüsselungsverfahren verwendet. Für die Auswertung werden die Pseudonyme auf die Relation Kleiner-Gleich getestet. Daraus wird auf die Relation der zugrundeliegenden Klartextdaten geschlossen.

Eingabe : Nach Algorithmus 16 erzeugte Pseudonyme $p_i \in P(D)$ mit $p_i := (p_{i,1},\ldots,p_{i,m},pcs_i)$, k pseudonymisierte Clusterzentren $C = \{pc_1,\ldots,pc_k\}$ mit $pc_i := (pc_{i,1},\ldots,pc_{i,m},pcs_i)$.
Ausgabe : Elementzähler ec_j, $1 \leq j \leq k$, Zuordnung $Z := [z_1,\ldots,z_{|P(D)|}]$ mit $\left\{(z_i := (p_i, pc_j)) \mid p_i \in P(D), pc_j \in C\right\}$, pseudonymisierte aufsummierte Clusterzentren $\{sc_1,\ldots,sc_k\}$, mit $sc_j = (sc_{j,1},\ldots,sc_{j,n},dsc_j)$ für alle $1 \leq j \leq k$.

1 **for** $i = 1,\ldots,|P(D)|$ **do**
2 $encDist = []$;
3 $A = []$;
4 **for** $j = 1,\ldots,k$ **do**
5 **Analyst:**
6 Berechne die verschlüsselten Differenzwerte zwischen p_i und pc_j komponentenweise (Operator $\boxminus$):
7 $dist = (dist_{i,j_1},\ldots,dist_{i,j_m},cs_{i,j}) = ((p_{i_1} \boxminus pc_{j_1}),\ldots,(p_{i_m} \boxminus pc_{j_m}),(cs_i \boxminus cs_j))$;
8 $encDist.append(dist)$;
9 **end**
10 Sende *encDist* an Dekryptor;
11 **Dekryptor** berechnet nach Algorithmus 20 den Index ind des Vektors mit der kürzesten Distanz und sendet diesen an den Analyst;
12 A.append(ind);
13 $ec++$;
14 **Analyst** summiert das verschlüsselte Clusterzentrum auf:
15 $sc_{ind} = (sc_{ind_1} \boxplus p_{i_1},\ldots,sc_{ind_m} \boxplus p_{i_m},cs_{ind} \boxplus cs_i)$;
16 **end**
17 **return** A, $ec_1,\ldots,ec_k$, *Zuordnung der Daten zu den Clustern*;

Algorithmus 19 : Analyst: Auswertung k-Means-Clustering: Zuordnung der Daten zu den Clustern in einer k-Means-Iteration.

Für die Umsetzung der Verkettbarkeit bezüglich der Elementrelation wurde ein bereits in [91] vorgestelltes Pseudonymisierungsverfahren verwendet, das auf einem Bloom-Filter kombiniert mit einer deterministischen Hash-Funktion mit Salt basiert. Für die Auswertung wird eine Schnittmengenanfrage an den den Bloom-Filter verwaltenden Dekryptor gesendet. Dieser bestimmt die Elementrelation.

Anforderungsklasse Operation Für die Umsetzung der Operationen Addition (und entsprechend Durchschnittsberechnung) und Multiplikation werden partiell-additiv- bzw. partiell-multiplikativ-homomorphe, asymmetrische probabilistische Verschlüsselungsverfahren eingesetzt. Für die Auswertung führt der Analyst die homomorphe Operation auf den Pseudonymen durch. Für die Entschlüsselung des Berechnungsergebnisses sendet er eine Aufdeckungsanfrage an den Dekryptor. Diese gibt die Klartextdaten für erlaubte Aufdeckungsanfragen durch Entschlüsselung frei.

Eingabe : Pseudonymisierte Differenzvektoren $dist_i \in P(D)$ mit $dist_i := (dist_{i,1}, \ldots, dist_{i,m}, distcs_i)$, privater Schlüssel k_{priv}, Gewichte $(w_1, \ldots, w_m)$.
Ausgabe : Index j des kürzesten Differenzvektors.

```
1  index = 0;
2  distance = ∞;
3  for j ∈ {1,...,k} do
4  |  Entschlüssele dist_j zu δ_j = (δ_{i,1},...,δ_{i,n},δcs_i);
5  |  sq_eukl = Σ_{i=1}^{n} δ_{j,i}^2;
6  |  test = Σ_{i=1}^{n} δ_{j,i} · w_i;
7  |  if test == δcs_j then
8  |  |  if sq_eukl < distance then
9  |  |  |  distance = sq_eukl;
10 |  |  |  index = j;
11 |  |  end
12 |  |  else
13 |  |  |  Abbruch, da Abfrage nicht erlaubter Berechnung entspricht;
14 |  |  end
15 |  end
16 end
17 Sende index an Analyst;
18 return Neuberechnete Clusterzentren;
```

Algorithmus 20 : Dekryptor: Auswertung k-Means-Clustering: Zuordnung der Daten zu Cluster in einer k-Means-Iteration.

Anforderungsklasse Algorithmus Für die Nutzbarkeitsanforderung Algorithmus wurde in Kapitel 3.4.6 motiviert, dass von einer Betrachtung des auf der Pseudonymisierung auszuführenden Algorithmus als Abfolge der beinhalteten Operationen als einzelne Nutzbarkeitsanforderungen abgesehen wird. Eine entsprechende nutzbarkeitserhaltende Pseudonymisierung für jede der enthaltenen Operationen wäre zwar technisch umsetzbar, jedoch aus Gründen des Vertraulichkeitsschutzes im Allgemeinen nicht zu empfehlen. Um für einen Algorithmus eine geeignete Pseudonymisierung zu generieren, wird zunächst eine Privatsphäre respektierende Variante des Algorithmus ausgewählt. Dann wird die Pseudonymisierung passend zu dieser Variante erstellt. So soll sichergestellt werden, dass die Pseudonymisierung möglichst wenig mehr Nutzbarkeit bietet als für die Ausführung des Algorithmus erforderlich ist. Die Auswertung der Nutzbarkeit des Algorithmus erfolgt durch die Ausführung eines auf diesen Algorithmus abgestimmten Mehrparteienprotokolls. So soll sichergestellt werden, dass die Ausführung des Algorithmus keine unintendierte Aufdeckung von Klartextdaten zur Folge hat.

Zusätzliche Schutzmassnahmen Abschließend bleibt festzuhalten, dass die Auswertung von Pseudonymen auf den verarbeitenden Systemen die Implementierung von zusätzlichen, über die Pseudonymisierung der Klartextdaten hinausgehende Schutzmaßnahmen erfordert. Hierfür kommt eine Reihe sorgfältig abgewogener technisch-organisatorischer Schutzmaßnahmen infrage.

Eingabe : Nach Algorithmus 16 erzeugte Pseudonyme $p_i \in P(D)$ mit $p_i := (p_{i,1},\ldots,p_{i,n},pcs_i)$, k pseudonymisierte Clusterzentren $C^{alt} = \left\{pc_1^{alt},\ldots,pc_k^{alt}\right\}$ mit $pc_i^{alt} := (pc_{i,1}^{alt},\ldots,pc_{i,n}^{alt},pcs_i^{alt})$ Elementzähler ec_j, $1 \leq j \leq k$, und die Zuordnung der Pseudonyme zu den pseudonymisierten Clusterzentren $Z := [z_1,\ldots,z_{|P(D)|}]$ mit $\left\{z_i := (p_i,pc_j) \mid p_i \in P(D), pc_j \in C\right\}$.

Ausgabe : Neue Clusterzentren $C = \{pc_1,\ldots,pc_k\}$ mit $pc_i := (pc_{i,1},\ldots,pc_{i,n},pcs_i)$.

1 **for** $i = 1,\ldots,|P(D)|$ **do**
2 $A_i := j$ für $(p_i,pc_j) \in Z$;
3 **if** $ec_{A_i} > 0$ **then**
4 Setze ec auf ec_{A_i} und somit auf die Anzahl der Elemente des Clusters, dessen Clusterzentrum pc_j ist;
5 Berechne pseudonymisierte Clusterzentren $C_{A_i} = (\{pc_1,\ldots,pc_k\})$ mit $pc_i := ((pc_{A_i,1} + \frac{p_{i,1}}{ec}),\ldots,(pc_{A_i,n} + \frac{p_{i,n}}{ec}),(pcs_{A_i} + \frac{ps_i}{ec}))$;
6 **end**
7 **end**
8 **for** $j = 1,\ldots,k$ **do**
9 **if** $ec_j == 1$: **then**
10 $C_j := pc_j^{alt}$;
11 **end**
12 **end**
13 **return** *Neue Clusterzentren* $C = \{pc_1,\ldots,pc_k\}$ *mit* $pc_i := (pc_{i,1},\ldots,pc_{i,n},pcs_i)$;

Algorithmus 21 : Analyst: Auswertung k-Means-Clustering: Anpassung der Clusterzentren in einer k-Means-Iteration.

Ein Beispiel ist das Monitoring der verarbeiteten Schritte. So soll sichergestellt werden, dass die Verarbeitungsschritte denen für die Auswertung der intendierten Nutzbarkeiten entsprechen. Diese Maßnahmen sollen zu einer möglichst umfassenden Minderung des Restrisikos der Reidentifizierung beitragen. Jedoch muss beachtet werden, dass die bloße Existenz und Verarbeitung von aus den Klartextdaten entnommenen Nutzbarkeiten eine Verfügbarkeit von Information darstellt. Somit ist hier stets ein Restrisiko der Anwendbarkeit für die Durchführung einer Reidentifizierung gegeben.

Zusätzlicher Berechnungsaufwand durch Pseudonymisierung Die erarbeiteten Pseudonymisierungsverfahren bringen im Vergleich zur Verarbeitung von Klartextdaten einen nicht unerheblichen Mehraufwand mit. Dies ist bei der Auswertung zu beachten. Die im Rahmen der Forschung zu der vorliegenden Arbeit durchgeführten Messungen haben gezeigt, dass nach aktuellem Stand der Wissenschaft pseudonymisierte Daten zumindest für Offline-Analysen geeignet sind. Eine Nutzung für Echtzeitanalysen insbesondere großer Datenmengen ist eingeschränkt für Pseudonymisierungen einzelner Nutzbarkeiten möglich. Bei den im vorliegenden Kapitel gelisteten Beispielen für nutzbarkeitserhaltende Pseudonymisierungsverfahren zählen derzeit Pseudonymisierungen, die auf Hash-Funktionen, symmetrischen Verschlüsselungsverfahren und ordnungsoffenbarenden

Eingabe : Elementzähler mit aufsummierten Zentren der zugehörigen Cluster: $\left\{(ec_j, sc_j) \mid 1 \leq j \leq k\right\}$, öffentlicher Schlüssel k_{pub}, geheimer Schlüssel k_{priv}, Gewichte $w_1, \ldots, w_n$.
Ausgabe : Neue Clusterzentren $\left\{C_j \mid 1 \leq j \leq k\right\}$.

1 **for** $j \in \{1, \ldots, k\}$ **do**
2 Entschlüssele sc_j zu $\sigma_j = (\sigma_{j_1}, \ldots, \sigma_{j_n}, \sigma cs_j)$;
3 $test = \sum_{i=1}^{n} \sigma_{j_i} \cdot w_i$;
4 **if** $test \neq cs_j$ **then**
5 Abbruch, da Abfrage nicht erlaubter Berechnung entspricht;
6 **end**
7 **else**
8 $new = \sum_{i=1}^{n} \frac{\sigma_{j_i}}{ec_j} \cdot w_i$;
9 Pseudonymisiere neue Clusterzentren nach Algorithmus 16 zu $Enc(C_j)$: $Enc(C_j) = (Enc(\frac{\sigma_{j_1}}{ec_j}), \ldots, Enc(\frac{\sigma_{j_n}}{ec_j}), C_j cs_j)$;
10 **end**
11 **end**
12 Sende die aktualisierten Clusterzentren $Enc(C_j)$ an den Analyst;
13 **return** *Neuberechnete Clusterzentren*;

Algorithmus 22 : Dekryptor: Auswertung k-Means-Clustering: Anpassung der Cluster in einer k-Means-Iteration.

Verschlüsselungsverfahren basierend generiert wurden zu den Verfahren, die eine Auswertung der Nutzbarkeiten in Echtzeit erlauben. Hintergrund ist, dass die Auswertung der Nutzbarkeit auf den mit diesen Verfahren erzeugten Pseudonymen identisch mit der entsprechenden Auswertung auf den zugrundeliegenden Klartextdaten ist. Pseudonymisierungen, die auf asymmetrischen, insbesondere homomorphen Verschlüsselungsverfahren basierend generiert wurden, sind wegen des erhöhten Speicher- und Rechenzeitbedarfs zur Ausführungszeit bisher nicht für die Echtzeitanalyse geeignet.

6 Von der Anforderungsbeschreibung zur Umsetzung: Pseudonymisierungsstruktur und Übersetzungsregeln

In Kapitel 4 wurde beschrieben, wie in dem in der vorliegenden Arbeit vorgestellten Rahmenwerk mit der Beschreibungssprache `Util` Anforderungen an eine Pseudonymisierung als Nutzbarkeitspolitik formuliert werden können. Um aus der Beschreibung der Anforderungen maßgeschneiderte Pseudonyme erzeugen zu können, müssen die Anforderungen zunächst in geeignete Pseudonymisierungsverfahren übersetzt werden. Diese Verfahren wurden in Kapitel 5 beschrieben. In diesem Kapitel sollen Übersetzungsregeln erarbeitet werden, mit denen aus in `Util` formulierten Anforderungen automatisiert eine maßgeschneiderte Pseudonymisierung generiert werden kann, die diese Anforderungen erfüllt. Die in Kapitel 5 beschriebenen Verfahren generieren Pseudonyme in unterschiedlichen Repräsentationen und Parametrisierungen. So ist zum Beispiel für die Erzeugung der Pseudonyme nach Algorithmus 12 ein öffentlicher Schlüssel des Paillier-Verschlüsselungsverfahrens [119] erforderlich. Dieser ist identisch für alle durch eine Nutzbarkeitsanforderung adressierten Daten. Für die Erzeugung von aufdeckbaren Pseudonymen nach Algorithmus 1 ist die Generierung eines Salts für jedes in die Nutzbarkeit eingehende Datum erforderlich. Der Nutzbarkeitsanforderung Operation Multiplikation genügende Pseudonyme bestehen aus zwei Komponenten für jedes zugrundeliegende Klartextdatum. Der Nutzbarkeitsanforderung Algorithmus genügende Pseudonyme bestehen aus $m+1$ Komponenten für jeden zugrundeliegenden, aus m Klartextdaten $d_{i,j}$ bestehenden Datensatz d_i.
Auch die Zusatzinformation, die für die Auswertung der Nutzbarkeit erforderlich ist, unterscheidet sich in Abhängigkeit vom gewählten Verfahren. Einige Pseudonymisierungsverfahren generieren Pseudonyme, bei denen die Auswertung der Nutzbarkeit identisch mit der Verarbeitung der zugrundeliegenden Klartextdaten ist. Dies gilt z.B. für das im Algorithmus 5 beschriebene Auswertungsverfahren. Andere erfordern für die Auswertung der Nutzbarkeit ein im Vergleich zur Verarbeitung der Klartextdaten aufwendiges Verfahren oder die Ausführung eines Mehrparteienprotokolls mit Zugriff auf öffentliche oder geheime Schlüssel. Beispiele sind hier die in den Algorithmen 9 und 12 gelisteten Verfahren. Die verschiedenen Pseudonymisierungsverfahren unterscheiden sich also sowohl hinsichtlich des Erstellungs- als auch des Auswertungsprozesses.
Die nutzbarkeitserhaltenden Pseudonyme weisen weiterhin Personenbezug auf, wenn auch im Vergleich zu personenbezogenen Klartextdaten in einer risikomindernden Form. Zum weiteren Schutz der Pseudonyme werden für diese Vertraulichkeitsanforderungen formuliert. Diese werden durch die Umsetzung zusätzlicher Datenschutzmechanismen realisiert.

Um trotz der dargestellten Heterogenität in der Umsetzung der Pseudonymisierungsverfahren und der Datenschutzmechanismen eine automatisierte, maßgeschneiderte Umsetzung von Nutzbarkeits- und Vertraulichkeitsanforderungen in Pseudonymisierungen zu ermöglichen, wird in Kapitel 6.1 eine vereinheitlichende Pseudonymisierungsstruktur erarbeitet. Für die in dieser Arbeit vorgestellten Nutzbarkeits- und Vertraulichkeitsanforderungen werden Regeln in Kapitel 6.2 zur Übersetzung der in `Util` formulierten Anforderungen in die Pseudonymisierungsstruktur beschrieben. Die Pseudonymisierungsverfahren, die hierbei zum Einsatz kommen, wurden bereits in Kapitel 5 beschrieben.

6.1 Pseudonymisierungsstruktur

Nutzbarkeitserhaltende Pseudonymisierungen erhalten die Nutzbarkeiten, die für die intendierte Datenverarbeitung erforderlich sind. Hierbei wird für jede Nutzbarkeit und jedes für diese Nutzbarkeit erforderliche Klartextdatum ein Pseudonym erzeugt. Ein Pseudonym wird innerhalb der Pseudonymisierungsstruktur Utility-Tag genannt. Ein Utility-Tag $u_{nu}(d)$ repräsentiert eine einzelne Nutzbarkeit nu eines einzelnen Klartextdatums d in der gesamten zu erstellenden Pseudonymisierung $P(D)$. Alle Utility-Tags der Daten $D_{nu} \subseteq D$, die in der Nutzbarkeitsanforderung nu adressiert werden, bilden das Pseudonym p_{nu} der Nutzbarkeit nu. Sie sind Teil der Pseudonymisierung $P(D)$:

$$p_{nu} := \bigcup_{d \in D_{nu}} u_{nu}(d) \subseteq P(D).$$

Die Gesamtheit der Utility-Tags aller Nutzbarkeitsanforderungen ergibt die Pseudonymisierung $P(D)$ einer Datensammlung D.
Die Utility-Tags $u_{nu}(d)$ der von nu adressierten Daten d_i aus D_{nu} aus D einer Nutzbarkeitsanforderung können an unterschiedliche Vertraulichkeitsanforderungen $ver_i(D_{nu})$ gebunden werden. Damit kann die Verfügbarkeit der Nutzbarkeiten weiter reglementiert werden.
In Abhängigkeit der gewählten Nutzbarkeiten kann für die Verarbeitung der Utility-Tags die Verwendung von zusätzlichen Daten in Form von kryptographischen Schlüsseln oder Salts erforderlich sein. Diese zusätzlichen Daten werden für jedes Utility-Tag separat abgelegt und werden diesem eindeutig zugeordnet. Da die zusätzlichen Daten häufig zum Schutz der Vertraulichkeit besonders geschützt und getrennt gespeichert werden müssen, werden sie in dieser Arbeit in einer sogenannten Secrets-Struktur $secrets(u_{nu}(d))$ aufbewahrt. Die Gesamtheit der Secrets einer Nutzbarkeit nu ist die Menge aller Secrets der Utility-Tags der Daten D_{nu}, die durch die Nutzbarkeitsanforderung in der Nutzbarkeitspolitik adressiert wurden: $\bigcup_{d \in D_{nu}} secrets(u_{nu}(d))$.
Für die Erstellung und Verarbeitung einer nutzbarkeitsorientierten Pseudonymisierung einer Datenmenge wird also eine Pseudonymisierungsstruktur erstellt, die zum einen für jedes einzelne in die Datenverarbeitung eingehende Klartextdatum mindestens ein Utility-Tag enthält. Zum anderen werden für eine Nutzbarkeit die Utility-Tags aller Klartextdaten, die diese Nutzbarkeit repräsentieren, gebündelt gehalten. Dies erleichtert die Erkennung aller von einer Nutzbarkeitsanforderung betroffenen Klartextdaten.

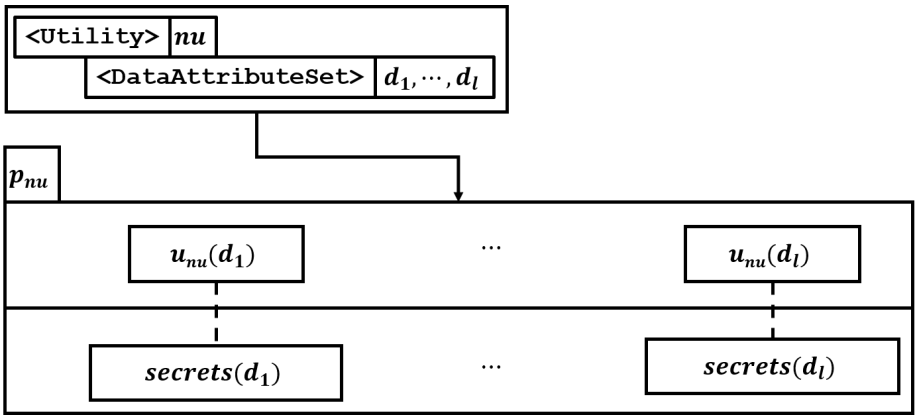

Abbildung 13: *Schematische Darstellung einer Sammlung von Utility-Tags mit den zugehörigen Secrets-Strukturen.*

Abbildung 13 enthält eine schematische Darstellung einer Utility-Tag-Sammlung mit Secrets-Struktur.
Bei der Verarbeitung der Pseudonymisierung werden verschiedene Nutzbarkeiten ausgewertet. Da der Auswertungsprozess einer Nutzbarkeit über die Verarbeitung einer Utility-Tag-Sammlung und der zugehörigen Secrets-Strukturen als Teil einer Pseudonymisierung eine Verarbeitung personenbezogener Daten darstellt, werden die Utility-Tag-Sammlungen zusätzlich geschützt. Dies erfolgt durch die Umsetzung von Vertraulichkeitsanforderungen. Dies umfasst zum einen die impliziten, beim Entwurf der Pseudonymisierungsstruktur berücksichtigten Vertraulichkeitsanforderungen. Zum anderen beinhaltet es die expliziten, in `Util` obligatorisch zu formulierenden Vertraulichkeitsanforderungen.

6.2 Regeln zur Übersetzung der Anforderungen in eine Pseudonymisierung

In diesem Abschnitt wird erarbeitet, wie aus in `Util` formulierten Anforderungen unter Nutzung der Pseudonymisierungsstruktur geeignete Pseudonymisierungen abgeleitet werden können. Für jede Gruppe von Anforderungen werden exemplarisch Übersetzungsregeln am Beispiel einer Nutzbarkeitsanforderungen der Anforderungsklasse beschrieben. Diese sollen eine automatisierte Erstellung von nutzbarkeitserhaltenden Pseudonymisierungen ermöglichen. Zunächst wird dafür in Abschnitt 6.2.1 die grundlegende Struktur einer Übersetzungsregel beschrieben. Diese Struktur umfasst die grundlegende Übersetzung von Nutzbarkeits- und Vertraulichkeitsanforderungen in eine Pseudonymisierungsstruktur. In Abschnitt 6.2.2 wird die Umsetzung der Übersetzungsregeln für ausgewählte Ausprägungen von Nutzbarkeitsanforderungen beschrieben. Die Umsetzung der Übersetzungsregeln für Vertraulichkeitsanforderungen erfolgt in Abschnitt 6.2.3.

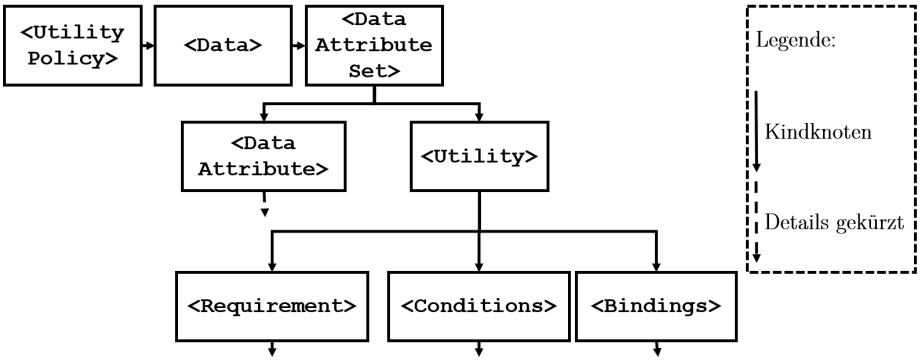

Abbildung 14: *Struktur einer* `Util`*-Politik.*

6.2.1 Grundlegende Struktur einer Übersetzungsregel

Wie in Kapitel 4 beschrieben, werden die Anforderungen an zu pseudonymisierende Daten nach Nutzbarkeiten gebündelt formuliert. Zu jeder Nutzbarkeitsanforderung werden also die betreffenden Klartextdaten d_j aus der Datensammlung D und die Vertraulichkeitsanforderungen formuliert, die bei der Umsetzung der Nutzbarkeitsanforderung beachtet werden müssen. Die diesem Ansatz folgende grundlegende Struktur einer Nutzbarkeitspolitik ist in Abbildung 14 skizziert. Das Element `<DataAttributeSet>` einer Nutzbarkeitsanforderung enthält dabei die mit dieser Nutzbarkeit zu pseudonymisierenden Klartextdaten d_j aus der Teilmenge D_i von D. Abbildung 18 zeigt eine schematische Darstellung der Umsetzung von nutzbarkeitserhaltender Pseudonymisierung durch die Anwendung von Übersetzungsregeln auf Anforderungen einer Nutzbarkeitspolitik.

Definition 71: Übersetzungsregeln
Eine Menge von Übersetzungsregeln

$$Transl_{Pol(D)}^{P(D)} := \bigcup_{i=1}^{k} \{nu_i(D_i) \mapsto (pseud_{nu_i}(D_i), sec_{nu_i}(D_i))\}, D_i \subseteq D\}$$

von einer Nutzbarkeitspolitik $Pol(D)$ mit k Nutzbarkeitsanforderungen über einer Datenmenge D in eine Pseudonymisierung $P(D)$ dieser Datenmenge ist definiert als eine Menge von Abbildungen, die nacheinander alle in $Pol(D)$ enthaltenen, als `type`-Wert eines `<Utility>`-Kindknoten `<Requirement>` definierten Nutzbarkeitsanforderungen $nu_i(D_i)$, $i \in 1,\ldots,k$ adressieren. Hierbei ist D_i die Menge der Daten, die im Elternknoten `<DataAttributeSet>` für den `<Utility>`-Kindknoten definiert sind. Jede dieser adressierten Anforderungen wird nacheinander in geeignete, nutzbarkeitserhaltende Pseudonymisierungsverfahren $pseud_{nu_i}(D_i)$ übersetzt. Die Pseudonymisierungsverfahren werden entsprechend der Sicherheits- und Nutzbarkeitsanforderungen geeignet parametrisiert. Für jedes Pseudonymisierungsverfahren $pseud_{nu_i}(D_i)$ werden daher geeignete Secrets $sec_{nu_i}(D_i)$ festgelegt. Diese enthalten die für die Auswertung der Nutzbarkeit erforderliche Zusatzinformation.

Tabelle 3: *Übersetzungsregel für die Anforderungsklasse Aufdeckbarkeit.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Disclosability	-	$CBC_s(AES-256_k(\cdot))$	s, k

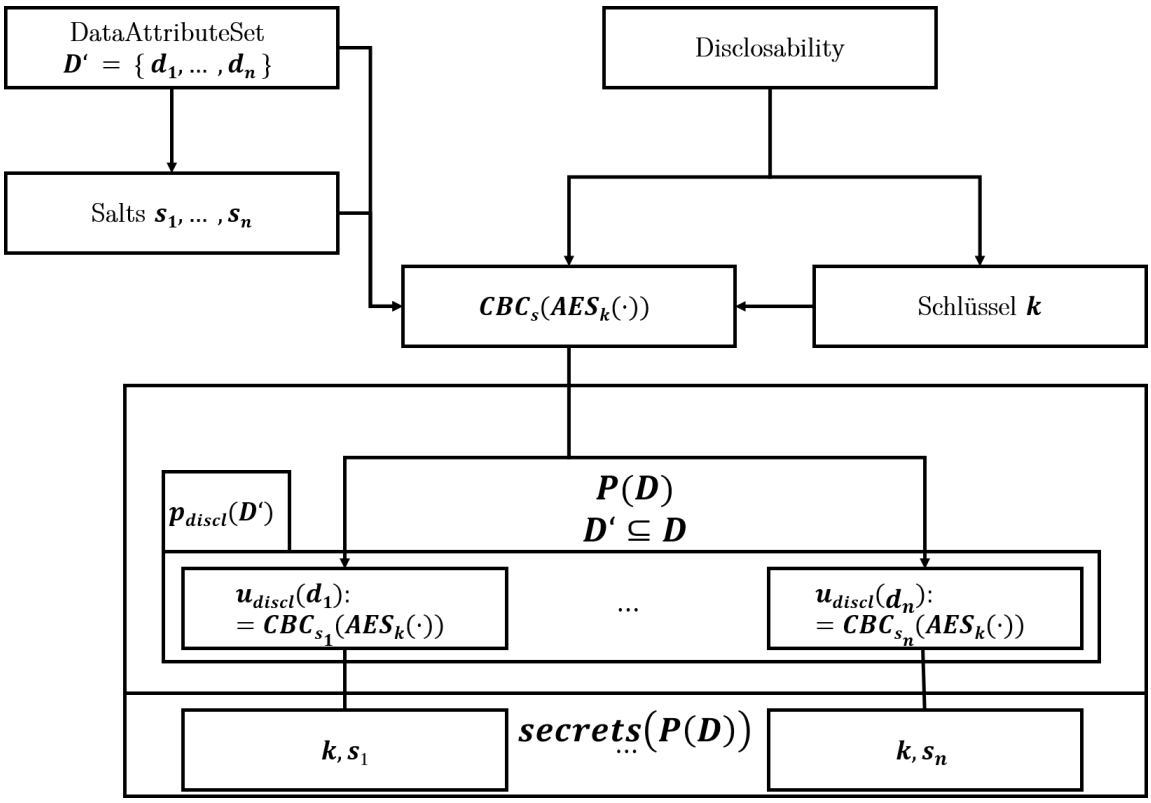

Abbildung 15: *Schematische Darstellung der Umsetzung der Übersetzungsregel für die Nutzbarkeitsanforderung der Aufdeckbarkeit.*

6.2.2 Nutzbarkeitsanforderungen

Die in Kapitel 3 beschriebenen Anforderungsklassen umfassen die Nutzbarkeitsanforderungen Aufdeckbarkeit, Verkettbarkeit bzgl. einer Relation, Operation und Algorithmus. Im Folgenden wird für jede dieser Anforderungsklassen der Nutzbarkeitsanforderungen für exemplarische Ausprägungen die Umsetzung einer geeigneten Übersetzungsregel beschrieben.

Anforderungsklasse Aufdeckbarkeit

Damit ein Klartextdatum d in einer Pseudonymisierung $P(D)$ der Nutzbarkeitsanforderung der Aufdeckbarkeit genügt, muss ein Utility-Tag $u_{discl}(d)$ aus $P(D)$ und eine zugehörige Secrets-Struktur nach der in Tabelle 3 beschriebenen Übersetzungsregel erzeugt werden. Das zugehörige Pseudonymisierungsverfahren wird in Algorithmus 1 beschrieben. In Abbildung 15 ist die Umsetzung der Übersetzungsregel skizziert. Für die Auswertung dieser Nutzbarkeitsanforderung wird Algorith-

Tabelle 4: *Übersetzungsregel für Nutzbarkeitsanforderung der Pseudonym-Pseudonym-Verkettbarkeit bezüglich der Gleichheitsrelation.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Linkability	Relation =	$SHA-3(\cdot \oplus s)$	$\emptyset$

Tabelle 5: *Übersetzungsregel für Nutzbarkeitsanforderung der Pseudonym-Klartext-Verkettbarkeit bezüglich der Gleichheitsrelation.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Linkability	Relation =	$SHA-3(\cdot \oplus s)$	s

mus 2 auf die nach der Übersetzungsregel erzeugten Utility-Tags angewendet. Die Parameter des Algorithmus sind der Salt s als Initialisierungsvektor des Betriebsmodus der Blockchiffre und der AES-Schlüssel k. Sie werden der zugehörigen Secrets-Struktur entnommen.

Anforderungsklasse Verkettbarkeit

Damit Klartextdaten $d_i \in D_{link,r}$ aus D durch ihre Repräsentierung innerhalb einer Pseudonymisierung $P(D)$ miteinander bezüglich einer Relation r verkettet werden können, muss die Pseudonymisierung Utility-Tags $u_{link,r}(d_i) \in P(D)$ enthalten, für die gilt

$$\exists\, link_r : P(D) \to r \text{ mit } link_r(u(d_i, d_j)) = r(d_i, d_j) \text{ für alle } d_i, d_j \in D_{link,r}.$$

Es muss also eine Abbildung $link_r$ geben, die die Verkettung der Klartextdaten unter Verwendung der zugehörigen Utility-Tags und ohne direkte Verarbeitung der Klartextdaten erlaubt.

Ist in der Nutzbarkeitspolitik eine Nutzbarkeitsanforderung `<Utility><Requirement type = "Linkability">` deklariert, wird zunächst der Kindknoten `<Relation>` ausgewertet. Die dort angegebene Relation wird in ein die Relation erhaltendes Pseudonymisierungsverfahren übersetzt.

Anforderung: Relation Gleichheit Damit ein Klartextdatum d in einer Pseudonymisierung $P(D)$ der Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Relation Gleichheit[1] genügt, müssen ein Utility-Tag $u_{link,=}(d)$ aus $P(D)$ und eine zugehörige Secrets-Struktur nach der in Tabelle 4 beschriebenen Übersetzungsregel erzeugt werden. Enthält der `<Utility><Requirement type = "Linkability">`-Kindknoten `<Relation>` den Wert `equality`, so wird dieser in das in Algorithmus 4 beschriebene Pseudonymisierungsverfahren übersetzt. Dieses erzeugt deterministisch aus den unter `<Pseudonyms>` mit `<SetBySetID>` referenzierten Klartextdaten eindeutige Pseudonyme. Die Übersetzungsregel ist für die Pseudonym-Pseudonym-Verkettbarkeit in Tabelle 4 und für die Pseudonym-Klartext-Verkettbarkeit in Tabelle 5 gelistet.

Die so erzeugten Utility-Tags können zur Verarbeitungszeit mit dem Algorithmus 5 bezüglich der Gleichheit der zugrundeliegenden Klartextdaten verkettet werden. Hierfür ist kein Zugriff auf zusätzliche Daten erforderlich. Daher ist die zugehörige Secrets-Struktur leer.

[1] Für die Relation r wird das Gleichheitszeichen $=$ gesetzt.

Tabelle 6: *Übersetzungsregel für Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Kleiner-Gleich-Relation.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Linkability	Relation $\leq$	$ORE_k(\cdot)$	$\emptyset$

Tabelle 7: *Übersetzungsregel für Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Kleiner-Gleich-Relation.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Linkability	Relation $\leq$	$ORE_k(\cdot)$	k

Tabelle 8: *Übersetzungsregel für Nutzbarkeitsanforderung der Pseudonym-Pseudonym-Verkettbarkeit bezüglich der Elementrelation.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Linkability	Relation element-of	$\mathcal{BF}_{fp,mf}$ $.Insert(SHA-3(\cdot \oplus s))$	fp, mf

Anforderung: Relation Kleiner-Gleich Damit ein Klartextdatum d in einer Pseudonymisierung $P(D)$ der Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Relation Kleiner-Gleich[2] genügt, müssen ein Utility-Tag $u_{link,\leq}(d) \in P(D)$ und eine zugehörige Secrets-Struktur nach der in Tabelle 6 beschriebenen Übersetzungsregel erzeugt werden. Das zugehörige Pseudonymisierungsverfahren wird in Algorithmus 7 beschrieben. Enthält der `<Utility><Requirement type = "Linkability">`-Kindknoten `<Relation>` den Wert `less-equal`, so wird dieser in das in Algorithmus 7 beschriebene Pseudonymisierungsverfahren übersetzt. Dieses erzeugt aus den unter `<Pseudonyms>` mit `<SetBySetID>` referenzierten Klartextdaten Pseudonyme, die bis auf eine geringe Ungenauigkeit ordnungserhaltend sind [38]. Die Übersetzungsregel für die Pseudonym-Pseudonym-Verkettbarkeit ist in Tabelle 6 gelistet. Die entsprechende Regel für die Pseudonym-Klartext-Verkettbarkeit ist in Tabelle 7 aufgeführt. Die so erzeugten Utility-Tags können zur Verarbeitungszeit mit dem Algorithmus 8 bezüglich der Kleiner-Gleich-Relation der zugrundeliegenden Klartextdaten verkettet werden. Hierfür ist kein Zugriff auf zusätzliche Daten erforderlich. Daher ist auch hier die zugehörige Secrets-Struktur leer.

Anforderung: Elementrelation Damit ein Klartextdatum d in einer Pseudonymisierung $P(D)$ der Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Elementrelation genügt, müssen ein Utility-Tag $u_{link,\in}(d)$ aus $P(D)$ und eine zugehörige Secrets-Struktur nach der in Tabelle 8 beschriebenen Übersetzungsregel erzeugt werden. Das zugehörige Pseudonymisierungsverfahren wird in Algorithmus 9 beschrieben. Enthält der `<Utility><Requirement type = "Linkability">`-Kindknoten `<Relation>` den Wert `element-of`, so wird dieser in das in Algorithmus 4 beschriebene Pseudonymisierungsverfahren übersetzt. Dieses erzeugt deterministisch aus den unter `<Pseudonyms>` mit `<SetBySetID>` referenzierten Klartextdaten Pseudonyme, die eine Aussage über die Elementrelation mit hoher Wahrscheinlichkeit erlauben. Die Übersetzungsregeln für die Pseudonym-Pseudonym- bzw. Pseudonym-Klartext-Verkettbarkeit sind in Tabelle 8 bzw. Tabelle 9 gelistet. Die so erzeugten

[2] Für die Relation r wird das Kleiner-Gleich-Zeichen $\leq$ gesetzt.

Tabelle 9: *Übersetzungsregel für Nutzbarkeitsanforderung der Pseudonym-Klartext-Verkettbarkeit bezüglich der Elementrelation.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Linkability	Relation element-of	$\mathcal{BF}_{fp,mf}$ $.Insert(SHA-3(\cdot \oplus s))$	fp, mf, s

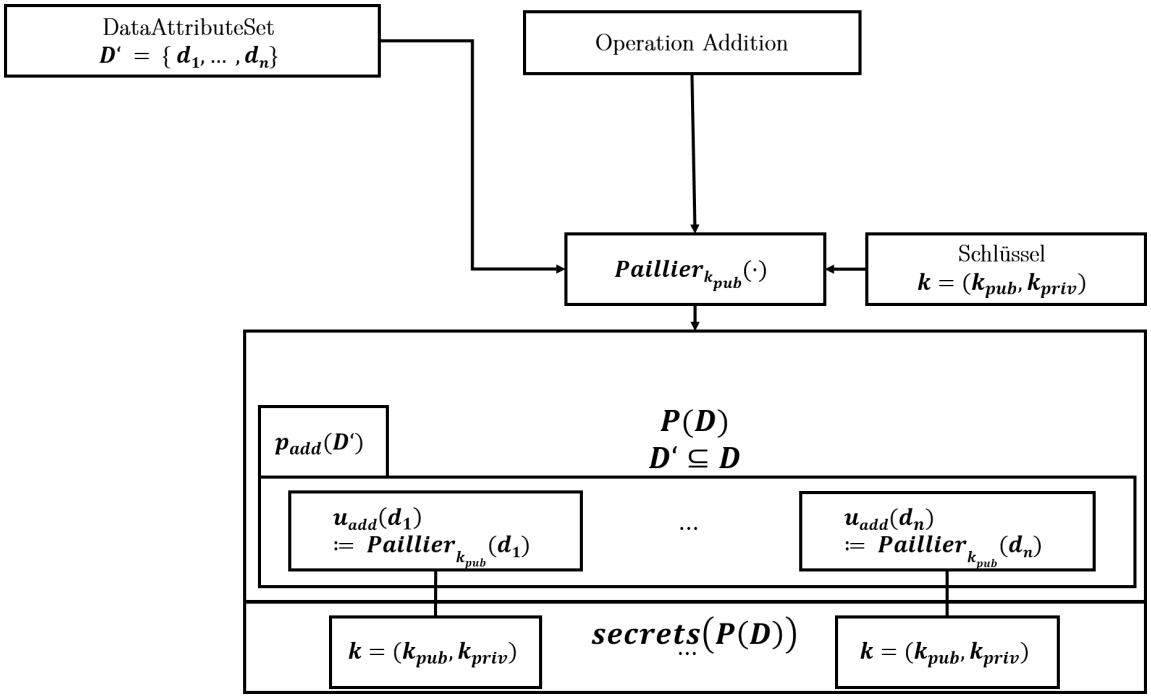

Abbildung 16: *Schematische Darstellung der Umsetzung der Übersetzungsregel für die Nutzbarkeitsanforderung der Operation Addition.*

Utility-Tags können zur Verarbeitungszeit mit dem Algorithmus 10 bezüglich der Kleiner-Gleich-Relation der zugrundeliegenden Klartextdaten verkettet werden. Hierfür ist jedoch der Zugriff auf zusätzliche Daten erforderlich. Diese beinhalten die gewünschte Falsch-Positiv-Rate fp und die gewünschte maximal Befüllungsrate mf des Bloom-Filters und für die Pseudonym-Klartext-Verkettbarkeit den Salt s der Hashfunktion. Diese werden in die Secrets-Struktur eingefügt.

Anforderungsklasse Operation

Damit auf Klartextdaten $d \in D$ in einer Pseudonymisierung $P(D)$ die Nutzbarkeitsanforderung einer Operation op umgesetzt werden kann, werden die Utility-Tags $u_{op}(d)$ aus $P(D)$ für diese Daten mittels homomorpher Verschlüsselung erzeugt. Die Parameter der Verfahren werden in die Secrets-Struktur eingefügt. Im Folgenden werden die Übersetzungsregeln für die Operationen Addition und Multiplikation beschrieben.

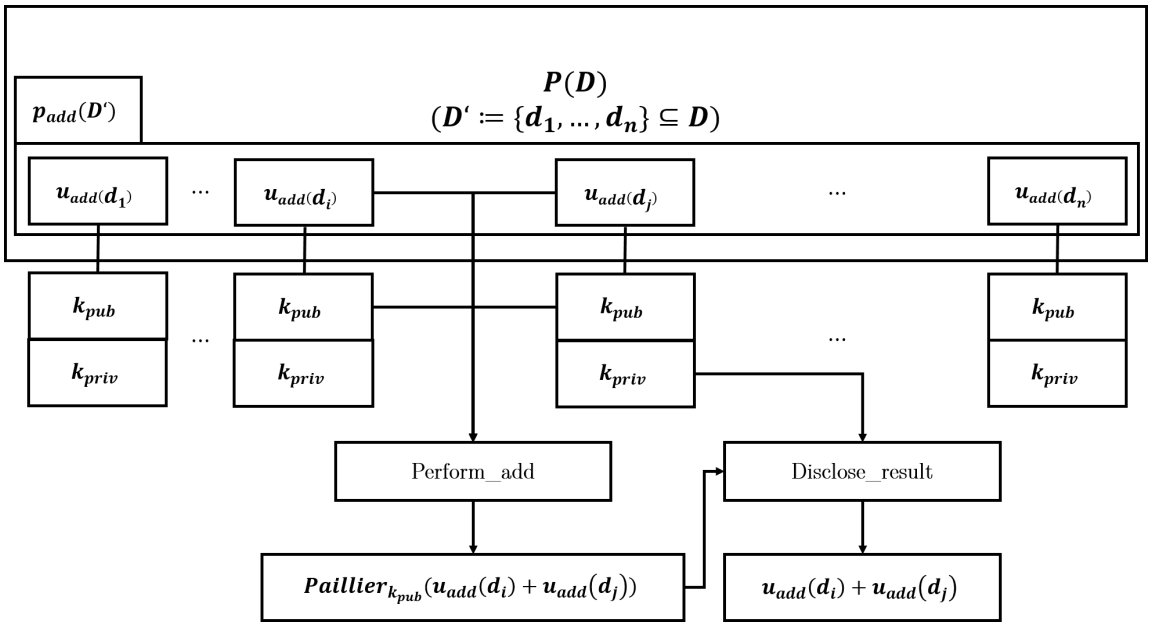

Abbildung 17: *Auswertung der Nutzbarkeit Addition auf den Utility-Tags.*

Tabelle 10: *Übersetzungsregel für die Nutzbarkeitsanforderung der Operation Addition.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Operation	Addition	$Paillier_{k_{pub}}(\cdot)$	k_{pub}, k_{priv}

Anforderung Addition Enthält der `<Utility><Requirement type = “Operation”>`-Kindknoten `<Type>` den Wert `Addition`, so wird dieser in das in Algorithmus 11 beschriebene Pseudonymisierungsverfahren übersetzt. Dieses erzeugt probabilistisch aus den unter `<DataSetAttributes>` referenzierten Klartextdaten einmalige Pseudonyme als Utility-Tags $u_{add}(d) \in P(D)$. Die Übersetzungsregel ist in Tabelle 10 gelistet. Die so erzeugten Utility-Tags können zur Verarbeitungszeit mit dem Algorithmus 12 mit anderen Pseudonymen so verknüpft werden, dass aus dem Ergebnis die Summe der den Pseudonymen zugrundeliegenden Klartextdaten entnommen werden kann. Hierfür ist jedoch der Zugriff auf zusätzliche Daten erforderlich. Diese sind der öffentliche und der private Paillier-Schlüssel. Sie werden daher in die Secrets-Struktur eingefügt. Eine schematische Darstellung der Umsetzung der Übersetzungsregeln befindet sich in Abbildung 16. In Abbildung 17 wird die Auswertung der Addition auf Utility-Tags schematisch dargestellt.

Ausprägung Multiplikation Enthält der `<Utility><Requirement type = “Operation”>`-Kindknoten `<type>` den Wert `Multiplication`, so wird dieser in das in Algorithmus 13 beschriebene Pseudonymisierungsverfahren übersetzt. Dieses erzeugt probabilistisch aus den unter `<DataSetAttributes>` referenzierten Klartextdaten einmalige Pseudonyme als Utility-Tags $u_{mult}(d) \in P(D)$. Die Übersetzungsregel ist in Tabelle 11 gelistet. Die so erzeugten Utility-Tags können zur Verarbeitungszeit mit dem Algorithmus 14 mit anderen Pseudonymen so verknüpft werden, dass aus dem Ergebnis das Produkt der den Pseudonymen zugrundeliegenden

Tabelle 11: *Übersetzungsregel für Nutzbarkeitsanforderung der Operation Multiplikation.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Operation	Multiplication	$Elgamal_{k_{pub}}(\cdot)$	k_{pub}, k_{priv}

Tabelle 12: *Übersetzungsregel für Nutzbarkeitsanforderung des Algorithmus k-Means.*

Requirement type	Ausprägung	Pseudonymisierungsverfahren	Secrets-Struktur
Algorithm	k-Means	$Paillier_{k_{pub}}(\cdot)$, Prüfsumme	$k_{pub}, k_{priv}, (w_1, \ldots, w_n)$

Klartextdaten entnommen werden kann. Hierfür ist jedoch der Zugriff auf zusätzliche Daten erforderlich. Diese sind der öffentliche und der private Elgamal-Schlüssel. Sie werden daher in die Secrets-Struktur eingefügt.

Anforderungsklasse Algorithmus

Damit eine PET-Variante eines Algorithmus *alg* auf pseudonymisierten Daten $P(D)$ ausgeführt werden kann, müssen entsprechend aufbereitete Utility-Tags $u_{alg}(d)$ für alle durch die Nutzbarkeitsanforderung adressierten Daten in $P(D)$ enthalten sein. Im Folgenden wird eine beispielhafte Übersetzungsregel für das *k*-Means-Verfahren beschrieben.

Anforderung *k*-Means Enthält der `<Utility><Requirement type = “Algorithm”>`-Kindknoten `<type>` den Wert `k-Means`, so wird dieser in das in Algorithmus 16 beschriebene Pseudonymisierungsverfahren übersetzt. Dieses erzeugt probabilistisch aus den unter `<DataSetAttributes>` referenzierten Klartextdaten einmalige Pseudonyme als Utility-Tags $u_{k-means}(d)$ aus $P(D)$. Die Übersetzungsregel ist in Tabelle 12 gelistet. Auf den so erzeugten Utility-Tags kann zur Verarbeitungszeit mit den Algorithmen aus Kapitel 18 ein Clustering berechnet werden. Hierfür ist jedoch der Zugriff auf zusätzliche Daten erforderlich. Diese sind der öffentliche und der private Paillier-Schlüssel, die Prüfsumme *ps* und die Gewichte $(w_1, \ldots, w_n)$ für m Datensätze mit jeweils n Attributwerten. Sie werden daher in die Secrets-Struktur eingefügt.

6.2.3 Vertraulichkeitsanforderungen

In Kapitel 6.2.2 wurden die Regeln zur Übersetzung von Nutzbarkeitsanforderungen in Utility-Tags beschrieben. Im Folgenden wird die Umsetzung der Vertraulichkeitsanforderungen beschrieben. Hierbei wird zwischen impliziten und expliziten Anforderungen unterschieden.

Implizite Anforderungen

Die Konstruktion der Übersetzungsregeln erfolgte auf Basis der in Kapitel 5 beschriebenen Pseudonymisierungsverfahren. Bei der Auswahl und dem Entwurf der Verfahren wurden implizite Vertraulichkeitsanforderungen berücksichtigt. Daher genügen die erzeugten Utility-Tags den impliziten Vertraulichkeitsanforderungen der Datenminimierung und der Beachtung des Standes der Wissenschaft und Technik. Die Anforderung der Beschränkung der Reidentifizierbarkeit Be-

troffener wird ebenfalls durch die Konstruktion der Pseudonymisierungsverfahren umgesetzt. Zusätzlich wird sie durchgesetzt, indem alle weiteren impliziten und expliziten Vertraulichkeitsanforderungen auch auf dem verarbeitenden System beachtet werden. Dazu gehört insbesondere die Trennung von Pseudonymisierung und Zusatzinformation. Diese wird durch das Design der Pseudonymisierungsstruktur als voneinander separierbare Utility-Tag-Struktur und Secrets-Struktur begünstigt.

Explizite Anforderungen

Nutzbarkeitserhaltende Utility-Tags ermöglichen einen intendierten Informationsabfluss. Zusätzlich ist weiterer, nicht intendierter Abfluss von Information möglich. Aktuell kann dieser zwar noch nicht quantifiziert werden. Jedoch können Beispiele wie das in Abschnitt 5 beschriebene erfasst werden. Wegen dieses Informationsabflusses ist ein zusätzlicher Schutz der Utility-Tags durch zusätzliches probabilistisches Verschlüsseln erforderlich. Dies erfolgt im Rahmen der expliziten Vertraulichkeitsanforderungen Rollen- bzw. Zweckbindung.

Rollenbindung Enthält der `<Utility>`-Kindknoten `<Bindings>` einen Kindknoten `<Binding>` mit dem `<Binding>`-Attribut `type` des Wertes `role`, so werden die für die entsprechende Nutzbarkeit erzeugten Utility-Tags zusätzlich probabilistisch verschlüsselt. Für jedes Utility-Tag wird hierbei ein frischer Salt s erzeugt. Für die Rolle r, deren Bezeichner dem Wert des `<Binding>`-Knotens entspricht, wird ein Schlüssel k_r erzeugt. Analog zu Algorithmus 1 wird der $AES-256$ im Cipherblock-Chaining-Modus mit den Parametern k_r und s aufgerufen. Das Ergebnis ist ein semantisch sicher erzeugtes Chiffrat. Im verarbeitenden System wird der Zugriff auf den Schlüssel k_r und den Salt s ausschließlich Subjekten der Rolle r ermöglicht. Nur diese können die so geschützten Utility-Tags entschlüsseln und entsprechend ihrer Nutzbarkeit verarbeiten. Die Durchsetzung der Rollenbindung muss im verarbeitenden System durch zusätzliche technisch-organisatorische Maßnahmen (TOM) erfolgen. Ein Beispiel ist die Implementierung von rollenbasierten Zugriffskontrollmodellen [57]. Sie sind in verschiedenen Betriebssystemen implementiert, u.a. FreeBSD [143] und SELinux [137]. Eine Herausforderung der Rollenbindung ist, dass Erkenntnisse aus der Datenverarbeitung ggf. vom Nutzer vorgehalten und innerhalb einer anderen Rolle weiter genutzt werden.

Zweckbindung Enthält der `<Utility>`-Kindknoten `<Bindings>` einen Kindknoten `<Binding>` mit dem `<Binding>`-Attribut `type` des Wertes `purpose`, so werden analog zur Rollenbindung die für die entsprechende Nutzbarkeit erzeugten Utility-Tags zusätzlich probabilistisch verschlüsselt. Für jedes Utility-Tag wird hierbei ein frischer Salt s erzeugt. Für den Zweck z, deren Bezeichner dem Wert des `<Binding>`-Knotens entspricht, wird ein Schlüssel k_z erzeugt. Analog zu Algorithmus 1 wird der $AES-256$ im Cipherblock-Chaining-Modus mit den Parametern k_z und s aufgerufen. Das Ergebnis ist ein semantisch sicher erzeugtes Chiffrat. Im verarbeitenden System wird der Zugriff auf den Schlüssel k_z und den Salt s ausschließlich Subjekten ermöglicht, die eine Verarbeitung im Sinne des Zwecks z nachweisen können. Nur diese können die so geschützten Utility-Tags entschlüsseln und entsprechend ihrer Nutzbarkeit verarbeiten. Auch hier muss die Durchsetzung der Zweckbindung im verarbeitenden System durch zusätzliche TOM erfolgen. Die besondere Her-

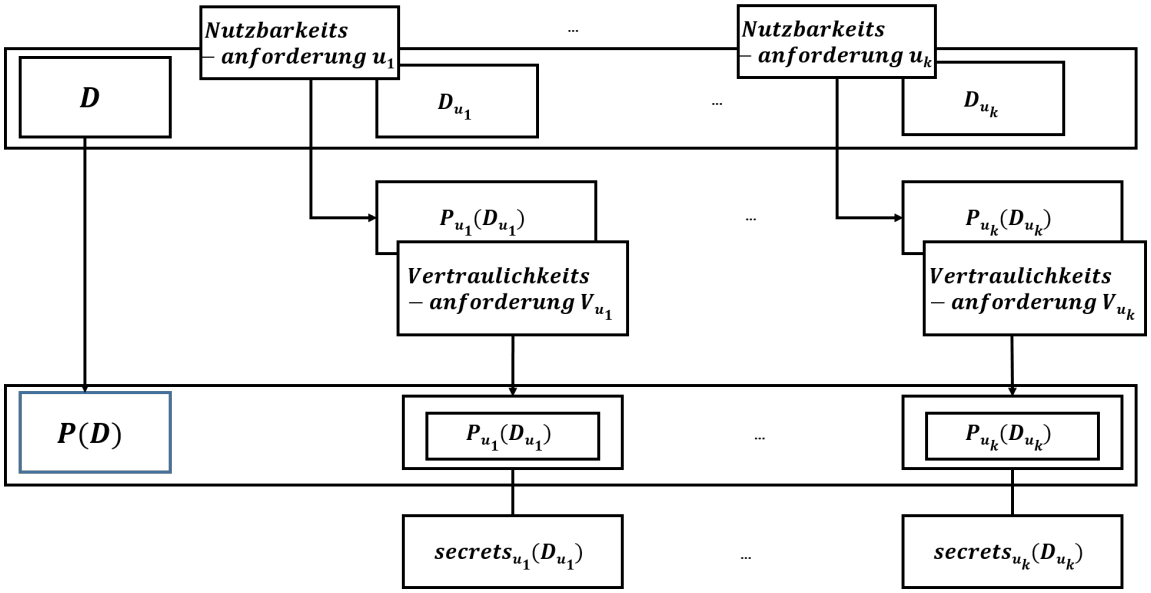

Abbildung 18: *Schematische Übersicht der Umsetzung der Vertraulichkeits- und Nutzbarkeitsanforderungen durch Übersetzungsregeln (durch Pfeile gekennzeichnet) in Pseudonymisierungen mit Utility-Tags und Secrets-Struktur.*

ausforderung ist, dass der Zweck zum Verarbeitungszeitpunkt nicht hinreichend sicher überprüft werden kann. Aus diesem und anderen Gründen scheitern bisher technische Konzepte wie das Digitale Rechte-Management [105]. Daher sind bei der Durchsetzung der Zweckbindung organisatorische Maßnahmen besonders zu beachten. Ein Beispiel ist die Erteilung eines Zugriffsrechts nach persönlicher Absprache und Befugniserteilung durch Verantwortliche. Dies kann umgesetzt werden, indem einem Subjekt der Zugriff auf den Schlüssel k_z und den Salt s nach persönlicher Absprache für eine geplante Dauer gewährt wird. Dies kann durch ein Monitoring und das Verhindern des Kopierens der Berechnungsergebnisse unterstützt werden. Jedoch kann auch hier nicht endgültig ausgeschlossen werden, dass Erkenntnisse aus einmal für einen Zweck verarbeiteten Daten nicht für einen anderen Zweck verwendet werden.

Begrenzung der Speicherungsdauer Die Speicherungsdauer von Utility-Tags einer Nutzbarkeit kann begrenzt werden. Diese Anforderung wird in einer `Util`-Politik umgesetzt, indem der `<Utility>`-Kindknoten `<Conditions>` mit dem Kindknoten `<Condition>` aufgerufen wird und das `<Condition>`-Attribut `type` mit dem Wert `expirationTime` belegt wird. Der Wert des `<Condition>`-Knotens entspricht einem Zeitstempel, zu dem das Utility-Tag gelöscht werden soll. Im verarbeitenden System müssen technisch-organisatorische Maßnahmen umgesetzt werden, die diese Anforderung umsetzen. Dies kann eine automatisierte Löschroutine sein, die alle Utility-Tags nach Erreichen des durch den Zeitstempel angegebenen Zeitpunkts vom System löscht.

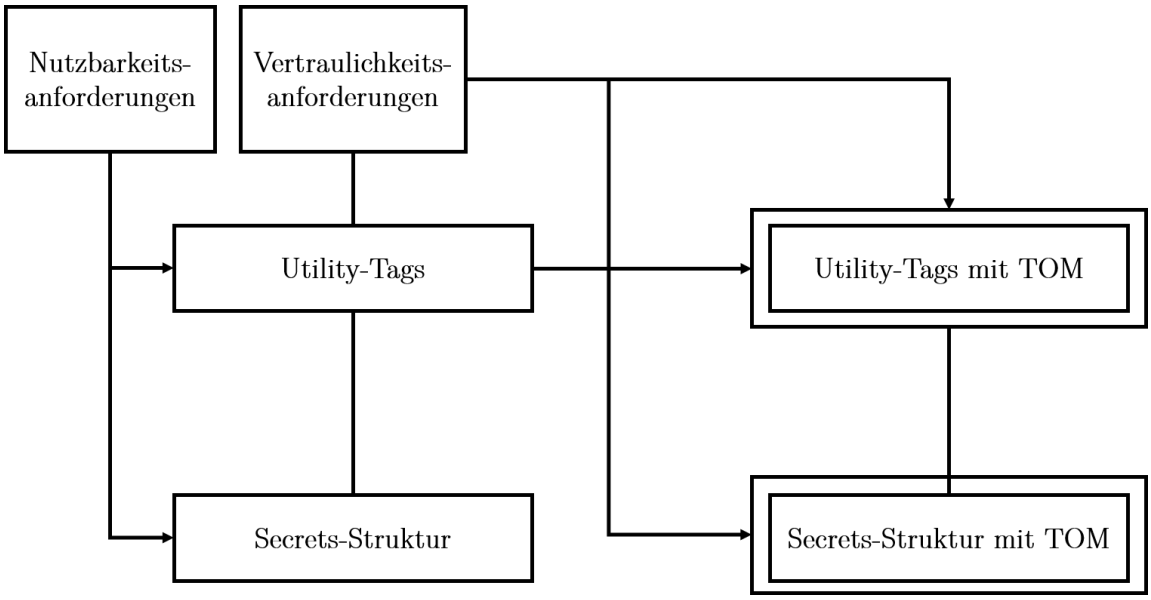

Abbildung 19: *Von den Nutzbarkeitsanforderungen zur Pseudonymisierung mit Utility-Tags und Secrets-Struktur. Auf dem ausführenden System werden ergänzende technisch-organisatorische Maßnahmen zum Schutz der Vertraulichkeit ergriffen.*

6.2.4 Fazit zu Übersetzungsregeln

Die im vorliegenden Kapitel beschriebenen Übersetzungsregeln ermöglichen eine automatisierte Ableitung von Pseudonymisierungen aus Anforderungen. Die Umsetzung erfolgt durch die Auswertung von in `Util` formulierten Nutzbarkeitsanforderungen. Aus diesen werden Utility-Tags und die zugehörigen Secrets-Strukturen generiert. Konstruktionsbedingt sind hierbei bereits implizite Vertraulichkeitsanforderungen eingeflossen. Weitere implizite und die expliziten Vertraulichkeitsanforderungen werden erst durch die in `Util` formulierten Vertraulichkeitsanforderungen als zusätzlich die Utility-Tags schützende probabilistische Verschlüsselung umgesetzt. Zusätzlich wird angenommen, dass das verarbeitende System weitere, die Vertraulichkeitsanforderungen umsetzende TOM implementiert und anwendet. Abbildung 19 veranschaulicht diesen Ansatz.

7 Anwendungsbeispiele

In Kapitel 2 wird ein Überblick über den in dieser Dissertation erarbeiteten Ansatz zur nutzbarkeitsorientierten Pseudonymisierung semistrukturierter textueller Daten gegeben. Die Komponenten des Ansatzes werden in den Kapiteln 3 bis 6 erarbeitet. Um die Praktikabilität des vorgestellten Rahmenwerks zu demonstrieren, werden in diesem Kapitel zwei Anwendungsbeispiele vorgestellt. Das erste Anwendungsbeispiel ist eine Privatsphäre respektierende Umfrage-Plattform. Diese wurde innerhalb studentischer Abschlussarbeiten erarbeitet, die im Rahmen der Forschungsarbeiten zu dieser Dissertation betreut wurden[1]. Im Gegensatz zu gängigen Umfrage-Plattformen werden die Daten hier nutzbarkeitserhaltend pseudonymisiert auf der Plattform verarbeitet. Dieses Beispiel wird in Abschnitt 7.1 beschrieben.
Das zweite Anwendungsbeispiel ist ein System zur Privatsphäre respektierenden Verwaltung der Warnung von Identitätsleaks Betroffener. Auch hier werden die verarbeiteten Daten nutzbarkeitserhaltend pseudonymisiert und verarbeitet. Dieses Beispiel wird in Abschnitt 7.2 beschrieben. Es wurde im Rahmen der Forschungsarbeit zu der vorliegenden Dissertation erarbeitet und veröffentlicht [91].

7.1 Anwendungsbeispiel Umfrage-Plattform

In einer Reihe von Forschungsgebieten werden im Rahmen von Experimenten personenbezogene Daten aus Umfragen ausgewertet. Da die Experimente die Beteiligung von Probanden erfordert, ist häufig vor der Durchführung das Einholen einer Einschätzung einer Ethikkommission erforderlich[2]. Begründet wird dies u.a. mit Gesetzen wie den Paragraphen §§ 40-42b des Arzneimittelgesetzes[3] oder §§ 20-23a des Medizinproduktegesetzes[4]. Der Kommission wird die Planung des Experiments dargelegt. Diese enthält eine möglichst detaillierte Beschreibung der geplanten Datenerhebung und -verarbeitung. Wird das Experiment durch die Ethikkommission freigegeben, kann es durchgeführt werden.
Zur Durchführung einer Umfrage greifen Forschende auch auf Online-Plattformen[5] zurück. Diese sollen die Erhebung und Verarbeitung der Umfragedaten erleichtern. Für ein Experiment erstellt der Forschende mindestens einen Fragebogen. Jeder Fragebogen besteht aus einer festgelegten Menge von Fragen und den möglichen Antworttypen für jede der Fragen. Mögliche Antworttypen sind numerische, kategorische, skalische (z.B. Likert-Skala) und Freitextantworten [58]. Somit sind

[1] siehe [107], [77] und [108]
[2] siehe z.B. die Ethikkommission der medizinischen Fakultät der Universität Bonn `https://ethik.meb.uni-bonn.de/`
[3] Siehe die entsprechenden Paragraphen des Arzneimittelgesetzes z.B. in [69]
[4] Siehe die entsprechenden Paragraphen des Medizinproduktegesetzes z.B. in [69]
[5] siehe z.B. SurveyMonkey: `https://www.surveymonkey.de`

vor der Freigabe des Fragebogens zur Beantwortung bereits die Datentypen und der Umfang der zu verarbeitenden Antworten bekannt. Sobald der Forschende die Umfrage eröffnet hat, können die Probanden den Fragebogen beantworten. Die Plattform speichert die Antworten und gibt die Sammlung der Antworten an den Forscher aus. Da ein beantworteter Fragebogen typischerweise von einem bestimmten Probanden beantwortet wurde und die enthaltenen Fragen häufig die Preisgabe persönlicher Umstände erfordert, handelt es sich bei den zu verarbeitenden Antworten meist um personenbezogene Daten. Diese unterliegen besonderem Schutz. Aktuell bekannte Plattformen [78] verarbeiten die Daten im Klartext. Somit gewähren diese Plattformen den Forschenden, aber auch jedem Individuum mit Zugriff auf die Plattform vollen Einblick in die gespeicherte personenbezogene Information. Dies können z.B. Angreifer sein, die an den personenbezogenen Daten von Teilnehmenden an Umfragen interessiert sind. Eine Reidentifizierung Betroffener ist somit häufig möglich. Auch können die Daten der Betroffenen ohne deren Einwilligung leicht zweckfremd verarbeitet werden. Potenzielle Teilnehmende an einer Umfragen können durch diese Risiken von einer Beantwortung eines Fragebogens absehen [43]. Eine Alternative, die die Daten Privatsphäre schützend verarbeitet, steht bisher aus. Das im Folgenden beschriebene erste Anwendungsbeispiel des in dieser Dissertation ausgearbeiteten Rahmenwerks für die nutzbarkeitserhaltende Datenpseudonymisierung soll die Möglichkeit einer Privatsphäre schützenden alternativen Umfrageplattform aufzeigen. Die Plattform ist im Rahmen der studentischen Arbeiten [77] und [108] entstanden.

7.1.1 Beschreibung der Anwendung

In diesem ersten Anwendungsbeispiel des Pseudonymisierungsansatzes wird eine Umfrageplattform mit nutzbarkeitserhaltender Datenpseudonymisierung dargestellt. Ausgangssituation ist die zu Beginn dieses Kapitels beschriebene Klartext-Umfrageplattform. Häufig ist die Erhaltung des Personenbezugs in den Antworten der Probanden für die Erkenntnisgewinnung erforderlich. Daher muss auf eine Anonymisierung der Daten verzichtet werden. Um das Risiko der Reidentifizierung Betroffener auch bei einem eventuellen Datenabfluss dennoch zu reduzieren, werden gezielt nutzbarkeitserhaltende Pseudonymisierungsverfahren eingesetzt.

Grundlegende Annahmen: Für den Nachweis der prinzipiellen Machbarkeit wird für die vorgeschlagene Umfrageplattform das folgende Szenario angenommen: (1) Die geplante Verarbeitung der Antworten auf die Fragebögen der auf der Plattform durchgeführten Umfragen ist zum Zeitpunkt der Erstellung der Umfrage im Detail bekannt. Dies umfasst die genaue Kenntnis der auf den Daten auszuführenden Operationen und Algorithmen. Ebenso ist bekannt, welche Information unter welcher Voraussetzung im Klartext aufgedeckt bzw. induziert werden soll. (2) Der Forschende in der Rolle des Analyst ist in der Lage, die aus der geplanten Verarbeitung der Antworten erforderlichen Nutzbarkeitsanforderungen abzuleiten. Dies wird dadurch unterstützt, dass die Nutzbarkeitsanforderungen in der Beschreibungssprache `Util` angegeben werden.
`Util` stellt für die expliziten Vertraulichkeitsanforderungen mandatorische Strukturen bereit. Diese Strukturen ermöglichen im Sinne des Vertraulichkeitsschutzes eine sinnvolle Instanziierung ohne Expertenkenntnisse in PET. Daher wird bei der Nutzung von `Util` angenommen, dass der

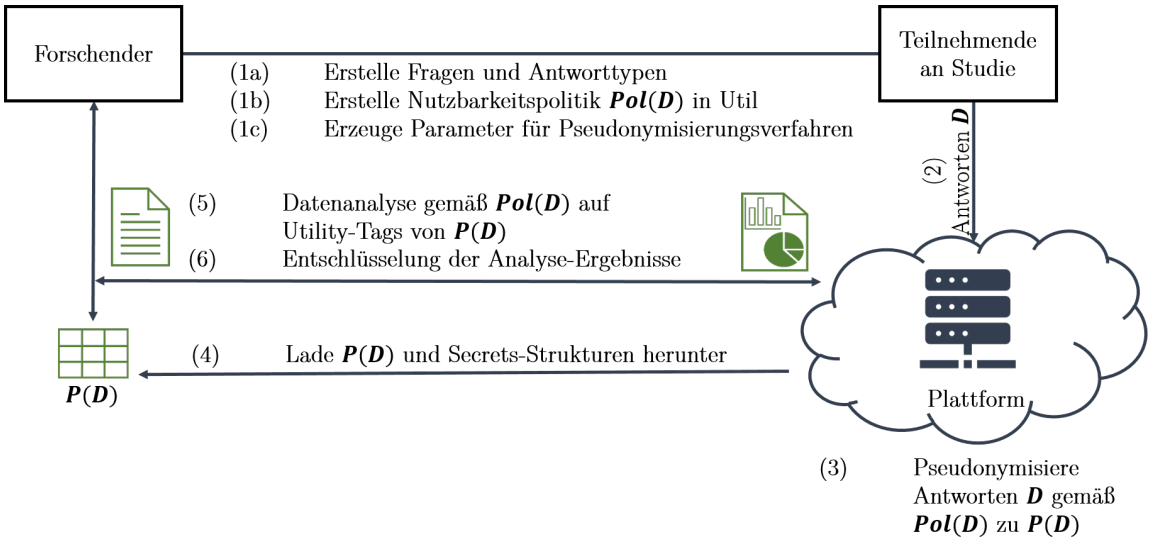

Abbildung 20: *Schematische Darstellung des Ablaufs der Analyse von Antworten einer Umfrage [108].*

Forschende auch ohne Expertenkenntnisse in der Lage ist, explizite Vertraulichkeitsanforderungen korrekt anzugeben. (3) Die Daten werden unmittelbar nach der Erfassung pseudonymisiert. Ist die Umfrage geschlossen, liegen die bereits pseudonymisierten Antworten als Pseudonymisierung $P(D)$ einer Datensammlung D vor.

Ablauf der Analyse von Antworten einer Umfrage Im ersten Schritt erstellt der Forschende einen Fragebogen. Diesem fügt er Fragen hinzu. Für jede der Fragen definiert er mögliche Antworttypen. Zur Auswahl stehen auf der Plattform numerische, kategorische und textuelle Antworttypen. Für jede der Antworten formuliert der Forschende die Nutzbarkeitsanforderungen an die Pseudonymisierung der Antworten in Form einer Nutzbarkeitspolitik $Pol(D)$ über die Datensammlung D aller möglichen Antworten d. Für die Formulierung der Politik nutzt der Forschende die Beschreibungssprache `Util`. Für jede der Nutzbarkeitsanforderungen formuliert er Vertraulichkeitsanforderungen. Diese umfassen mindestens die Rollen- oder Zweckbindung. So wird sichergestellt, dass die Daten nach der Erfassung in Form von nutzbarkeitserhaltenden Utility-Tags vorliegen, die probabilistisch verschlüsselt gespeichert werden. Davon ausgehend erzeugt die Plattform die Parameter für die geplante Datenpseudonymisierung.
Im zweiten Schritt erzeugen die Teilnehmer der Umfrage die Datensammlung D durch Beantwortung des Fragebogens. Im dritten Schritt wendet die Plattform die Datenpseudonymisierung auf D an und erzeugt somit $P(D)$. Das auf der Plattform integrierte Pseudonymisierungstool wendet nun die in Kapitel 6 beschriebenen Übersetzungsregeln automatisiert auf die in der Politik formulierten Anforderungen an. Das Ergebnis ist eine Pseudonymisierung der Datensammlung aller Antworten. Diese besteht aus Utility-Tags. Für jedes Utility-Tag existiert nun eine Secrets-Struktur, in der die für die Auswertung der Nutzbarkeit erforderliche Zusatzinformation gespeichert ist.
Um die Datenanalyse durchzuführen, fragt der Forschende im vierten Schritt die Pseudonymisierung von der Plattform ab. Die Plattform bereitet die pseudonymisierten Daten für das Senden an

den Forscher vor. Sie bereitet ebenso die Elemente der Secrets-Struktur, die für die Durchführung der Analyse erforderlich sind vor. Beides wird dem Forscher zur Verfügung gestellt. Dies beinhaltet explizit nicht die geheimen Schlüssel, die zur Auswertung der Aufdeckbarkeit bzw. zur Entschlüsselung von Berechnungsergebnissen erforderlich sind. Die Utility-Tags der Pseudonymisierung werden direkt nach ihrer Erstellung für die Zweck- oder Rollenbindung probabilistisch mit dem geheimen Schlüssel des Forschers verschlüsselt und diesem nach Abschluss der Umfrage bereitgestellt. Der Forschende entschlüsselt im fünften Schritt die empfangenen Daten zu $P(D)$ und führt die Analysen entsprechend der in $Pol(D)$ formulierten Anforderungen auf $P(D)$ durch. Je nach auszuwertenden Anforderungen sind die Ergebnisse der Analyse verschlüsselt. Um die Ergebnisse im sechsten Schritt entschlüsseln zu lassen, stellt der Forschende eine entsprechende Anfrage über ein hierfür von der Plattform bereitgestelltes Formular. Das Formular grenzt die Anfragemöglichkeiten in Abhängigkeit der Politik so ein, dass ein weiterer Informationsabfluss weiter reduziert wird. Die Ergebnisse der Berechnungen werden von der Plattform entschlüsselt und dem Forscher im Klartext zur Verfügung gestellt. Dieser Ablauf ist in Abbildung 20 schematisch dargestellt.

7.1.2 Nutzbarkeitsanforderungen und Evaluation

Für die Evaluierung der Plattform wurde exemplarisch eine Nutzbarkeitspolitik mit den Nutzbarkeitsanforderungen Verkettbarkeit bezüglich der Relation Gleichheit, Verkettbarkeit bezüglich der Relation Kleiner-Gleich, Operation Addition, Operation Multiplikation und Aufdeckbarkeit auf einer festen Datensammlung aus Antworten einer Umfrage mit der systeminternen *id* mit dem Wert 12345 betrachtet. Diese sollen zweckgebunden für eine Analyse A ausgewertet werden können. Dies ist entsprechend in dem Knoten `<Bindings>` definiert. Die Nutzbarkeitspolitik wurde in `Util` formuliert. Diese ist in Abschnitt 8.3 des Anhangs dieser Arbeit für alle Nutzbarkeitsanforderungen eines einzelnen Datenattributs `age` gelistet. Die Anforderungsformulierung der weiteren, in Tabelle 13 gelisteten Attribute ist identisch zu der gelisteten. Daher werden sie in der Darstellung der Politik im Anhang gekürzt. Für die Auswertung der Aufdeckbarkeit muss zusätzlich die Bedingung erfüllt werden, dass eine zuvor ausgewertete Operation Addition auf der gesamten Datensammlung eine Summe kleiner oder gleich dem Wert 0,1 ergibt. Dies ist als entsprechende `<Condition>` mit dem Typ `opResult` formuliert. Die Adressierung der auszuwertenden Operation erfolgt über den Kindknoten `<OperationID>`. Die von der Bedingung betroffenen zugrundeliegenden Klartextdaten werden über den Kindknoten `<SetID>` adressiert. Die als `<comparanceRelation>` formulierte Relation wird zum Vergleich des Berechnungsergebnisses mit dem in `<comparanceValue>` gespeicherten Wert verwendet.
Die `Util`-Politik wird der Plattform übergeben. Auf der Plattform wird die automatisierte Übersetzung in Pseudonymisierungsverfahren durch Anwendung der entsprechenden Übersetzungsregeln aus Kapitel 6 ausgeführt. Im Ergebnis wird aus einer ausgewählten Datensammlung automatisiert eine maßgeschneiderte nutzbarkeitserhaltende Pseudonymisierung generiert.

Tabelle 13: *Attribute der Datensätze für die Messung der Laufzeiten.*

Attributname	Beschreibung	Typ
first_name	Vorname	textuell
last_name	Nachname	textuell
postal_code	Postleitzahl	numerisch
location_city	Stadt	kategorisch
location_state	Bundesland	kategorisch
birthday	Geburtsdatum	numerisch
email	E-Mail-Adresse	textuell
phone	Telefonnummer	textuell
age	Alter	numerisch
gender	Geschlecht	kategorisch
religion	Religionszugehörigkeit	kategorisch
marital_status	Familienstand	kategorisch

Datensammlung

In der Abschlussarbeit [107] wurde eine Datensammlung entwickelt, die aus personenbezogenen Daten besteht. Sie wurde ähnlich den Zensusdaten modelliert und um weitere Attribute ergänzt. Jede Zeile der Datensammlung entspricht einem Datensatz mit Antworten eines fiktiven Teilnehmer einer Umfrage. Eine Zeile enthält entsprechend personenbezogene Attributwerte. Die Attribute der Datensätze sind in Tabelle 13 gelistet. Aus der in [107] beschriebenen Datensammlung wurden zufällig 1000 Datensätze gezogen. Diese wurden als Datensammlung für die Evaluierung verwendet.

Laufzeiten der Datenpseudonymisierung

Gemäß der `Util`-Politik 1 aus dem Anhang 8.3 wurde für jeden der in der Datensammlung vorhandenen Attributwerte jeweils ein Utility-Tag für die Nutzbarkeitsanforderungen Verkettbarkeit bezüglich der Relation Gleichheit, der Verkettbarkeit bezüglich der Relation Kleiner-Gleich, der Operation Addition und der Operation Multiplikation generiert. Die Erzeugung der Pseudonymisierung wurde auf einem Windows-10-System mit einer Intel-i5-3450-CPU mit 3,1 GHz und 24 GB RAM getestet. Das Backend der Umfrage-Plattform wurde in Python und die Pseudonymisierungsverfahren in Java implementiert. In Tabelle 14 sind die Laufzeiten für die Erzeugung der Pseudonymisierung als Gesamtheit der Utility-Tags mit Secrets-Struktur aus vorgegebenem Zusatzmaterial gelistet. Die Laufzeiten sind über jeweils 1000 Attributwerte gemittelt. Wie zu erwarten ist, ist die Ausführung der auf homomorphen Verschlüsselungsverfahren basierenden Pseudonymisierungsverfahren für die Operationen Addition und Multiplikationen sehr rechen- und daher am meisten zeitintensiv. Hintergrund sind die durchgeführten rechenintensiven Potenzberechnungen [119, 54]. Das auf der Nutzung von Pseudozufallszahlengeneratoren basierende Pseudonymisierungsverfahren für die Verkettbarkeit bezüglich der Relation Kleiner-Gleich ist deutlich weniger zeitintensiv. Dies deckt sich mit den Erkenntnissen von Bogatov et al. zur Evaluierung von ordnungsoffenbarenden Verschlüsselungsverfahren [24]. Das Pseudonymisierungsverfahren mit der geringsten Laufzeit ist jenes für die Verkettbarkeit bezüglich Gleichheit. Dies deckt sich mit den Angaben der NIST zur Performanz der zugrundeliegenden SHA-3-Hashfunktion [35].

Tabelle 14: *Mittelwerte der Laufzeiten der Erzeugung der Utility-Tags für die einzelnen Nutzbarkeiten.*

Nutzbarkeit	Erzeugung der Utility-Tags
Verkettbarkeit bzgl. $=$	0,059 s
Verkettbarkeit bzgl. $\leq$	0,156 s
Operation Addition	46,982 s
Operation Multiplikation	6,721 s

7.1.3 Fazit zur Umfrage-Plattform

Insgesamt kann festgehalten werden, dass mit der vorgestellten, die Privatsphäre respektierenden Umfrageplattform die folgenden Verbesserungen gegenüber herkömmlichen Plattformen erzielt wurden:

Verhinderung des Abfliessens sensibler Information Im Gegensatz zu bisherigen Ansätzen werden auf der Plattform keine Klartextdaten mehr gespeichert. Selbst zum Zeitpunkt der Durchführung der Datenanalyse liegen die Daten weder beim Analyst noch auf der Plattform im Klartext vor. Ein Angriff, durch den die gespeicherten Daten in die Hände Unbefugter geraten, hat so nicht mehr zur Folge, dass die Unbefugten Kenntnis über personenbezogene, möglicherweise sogar sensible Inhalte erlangen. Es besteht ein Restrisiko des Abfließens der Daten zum Zeitpunkt der Erstellung der Pseudonymisierung. Da jedes Datum unmittelbar nach seiner Erfassung auf der Plattform pseudonymisiert wird und die Daten dementsprechend nicht im Klartext vorgehalten werden, wird dieses Risiko verglichen mit der Situation bekannter Umfrageplattformen als gering erachtet.

Vertrauensfördernde Technik Die Daten liegen zu jedem Zeitpunkt pseudonymisiert auf der Plattform. Dies hat zur Folge, dass der Analyst zu keinem Zeitpunkt Zugriff auf die Klartextdaten erhält. Personen, die aus Sorge um den Schutz ihrer Privatsphäre nicht bereit sind, dem Analyst Auskunft über ihre persönlichen Verhältnisse zu geben, scheiden als Probanden bei der Nutzung herkömmlicher Plattformen aus. Das Wissen um den Umstand der nutzbarkeitserhaltenden Pseudonymisierung kann diese Personen zur Teilnahme an Umfragen ermutigen. Im Ergebnis erlangt der Analyst Aussagen über eine größere Population und damit ggf. aussagekräftigere Ergebnisse seiner Experimente.

Stärkung der Durchsetzung der Zweckbindung Die Daten werden maßgeschneidert für die Nutzbarkeiten einer bestimmten Analyse pseudonymisiert. Daher können sie für weitere Analysen allenfalls eingeschränkt genutzt werden. Zusätzlich zur Durchsetzung der Zweckbindung durch Verschlüsselung ist dies eine weitere, mittelbare Möglichkeit zur Stärkung der Zweckbindung der Datenverarbeitung.

Zusammenfassend kann festgehalten werden, dass mit der Umfrageplattform eine die Privatsphäre respektierende Alternative zu existierenden Plattformen erarbeitet wurde. Im Gegensatz zu

existierenden Ansätzen[6] liegen die Daten unmittelbar nach ihrer Erhebung zu keinem Zeitpunkt im Klartext vor. Entsprechend ist das Risiko eines unintendierten Abfließens personenbezogener, mitunter sensibler Daten deutlich verringert.

7.2 Anwendungsbeispiel Privacy-preserving Leakage Warning Management

Die personalisierte Nutzung von Online-Diensten geht vielfach einher mit der Erstellung und Nutzung von persönlichen Nutzerkonten. Dies geht mitunter mit der Weitergabe personenbezogener Daten an den Bereitstellenden des Online-Dienstes einher. Auch die Authentifikation an diesem geschieht wiederholt mit einer Kombination aus personenbezogenen Daten, z.B. einer E-Mail-Adresse und einem Passwort. Einige Nutzer haben folglich Nutzerkonten bei verschiedenen Online-Diensten, für die sie dieselben personenbezogenen Daten zur Erstellung, aber auch zur Authentifizierung an verschiedenen Online-Diensten nutzen [75].
Immer wieder kommt es in der Praxis vor, dass die von Online-Diensten gespeicherten personenbezogenen Daten der Nutzer vom verarbeitenden System abfließen (geleakte Daten) und in die Hände Dritter geraten [75, 103]. Dies kann für den einzelnen Nutzer schwerwiegende Folgen haben. Beispiele hierfür sind Verletzungen der Privatsphäre durch die Kenntnisnahme Dritter über die Nutzung eines bestimmten Dienstes oder auch Identitätsdiebstahl mit schwerwiegenden Folgen wie einem Kreditkartenbetrug [98]. Wegen der vielfältigen Nutzungsmöglichkeiten gesammelter geleakter Daten auch für illegale, für Kriminelle lukrative Aktivitäten werden diese Daten häufig im Darknet zum Download angeboten [102]. Kriminelle nutzen diese Möglichkeiten zur Weiterverfolgung ihrer Aktivitäten. Als Konsequenz streben Betreiber von Online-Diensten eine frühzeitige Warnung ihrer Nutzer an, wenn sogenannte Leaks zur Authentifizierung genutzt werden. Um zu erfahren, welche personenbezogenen Daten Leaks darstellen, müssen die Betreiber die Möglichkeit haben, die Daten ihrer Nutzer mit bereits bekannten Leaks abzugleichen. Auch nach einem Informationsabfluss unterliegen Leaks weiterhin den besonderen rechtlichen Bestimmungen zur Nutzung personenbezogener Daten[7]. Sie dürfen also nur unter besonderen Vorkehrungen erfasst und weiterverarbeitet werden. Dies beinhaltet auch die Nutzung der Daten zur Warnung der Nutzer. So soll der Nutzer in die Lage versetzt werden, zeitnah reaktiv Anmeldedaten wie das Passwort zu ändern. Wenn nun mehrere Online-Dienste ihre Nutzer im Falle von geleakten Daten warnen, so muss durch die häufige Wiederverwendung derselben Daten für die Authentifizierung an verschiedenen Diensten davon ausgegangen werden, dass dieselbe Person vielfach von verschiedenen Diensten bezüglich desselben Vorfalls gewarnt wird. Wenn man annimmt, dass Menschen bei Informationsüberflutung zum Ignorieren solcher Meldungen neigen, so ist eine Begrenzung der Anzahl der Warnungen sinnvoll. Dies erfordert jedoch eine Koordinierung mit entsprechendem Informationsaustausch zwischen verschiedenen Online-Diensten. Eine solche Koordinierung ist in Abbildung 21 skizziert.
Online-Diensten ist es aus rechtlichen Gründen häufig nicht gestattet, identifizierende personen-

[6] Siehe z.B. SurveyMonkey [78].
[7] Siehe Kapitel 2 der Arbeit von Malderle [103].

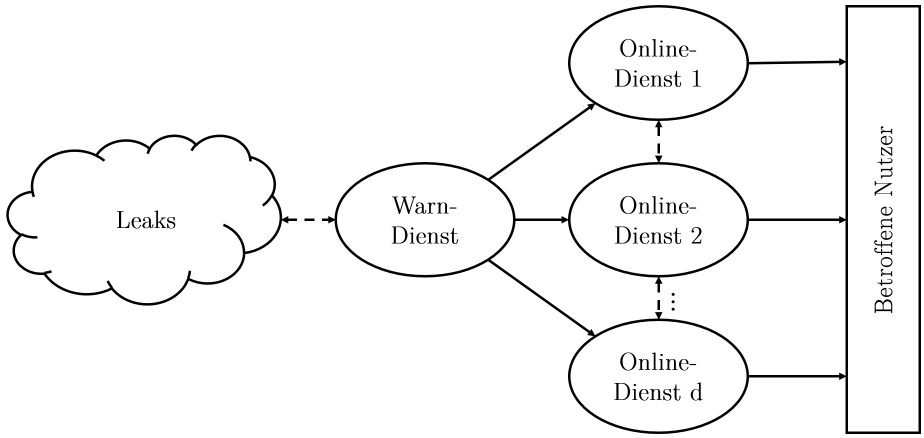

Abbildung 21: *Ein Netzwerk eines Warn-Verwaltungsdienstes mit mehreren Online-Diensten nach [91].*

bezogene Daten an weitere Online-Dienste herauszugeben [103]. Im Folgenden wird eine Lösung beschrieben, die durch den Einsatz von nutzbarkeitserhaltender Pseudonymisierung die Warnung von Nutzern von Online-Diensten ermöglicht, ohne diese bei einem Vorfall mehr als einmal zu warnen. Dies wird umgesetzt, indem Online-Dienste einen Warn-Verwaltungsdienst vorab abfragen können, ob ein Betroffener bereits gewarnt wurde. Gleichzeitig erlangen weder andere Online-Dienste noch der Warn-Verwaltungsdienst Kenntnis von der Identität der Betroffenen. Somit wird der Anforderung des Schutzes der Vertraulichkeit der Daten genügt.

7.2.1 Beschreibung der Anwendung

Online-Dienste nehmen an, dass ihre Nutzer Nutzerkonten bei verschiedenen Online-Diensten anlegen und dabei mehrfach dieselben Nutzerdaten verwenden [91, 103]. Wenn diese Daten in einem Leak gefunden werden, so sind entsprechend Nutzerkonten bei mehreren Online-Diensten gefährdet. Wenn die Dienste ihre Betroffenen entsprechend warnen wollen, wollen sie dennoch ein vielfaches Warnen vermeiden. Eine Lösung ist, wenn die Dienste einander mitteilen können, ob Betroffene eines Leaks bereits gewarnt wurden. Erhält ein Dienst Kenntnis über einen Leak, so kann er genau dann die Betroffenen warnen, wenn sie nicht bereits von anderen Diensten gewarnt wurden. Für diese Idee wurde das Warnungsnetzwerk entwickelt und von Kasem-Madani et al. in [91] beschrieben. Es ermöglicht Online-Diensten zu ermitteln, ob Betroffene bereits gewarnt wurden. Gleichzeitig wird die Identität der Betroffenen nicht weitergegeben.
Die für die Beschreibung der Anwendung relevanten Protokolle aus [91] werden in Anhang als Protokolle 23 (Vorbereitung), 24 (Datenaufbereitung) und 25 (Abfrage) gelistet.

7.2.2 Nutzbarkeitsanforderung und explizite Vertraulichkeitsanforderungen

Der Online-Dienst stellt die Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Elementrelation mit einer Menge von bereits gewarnten Betroffenen. Er möchte also ermitteln, ob ein

personenbezogenes Datum, z.B. eine E-Mail-Adresse, das in einer Menge von Leaks enthalten ist, bereits gewarnt wurde. Der Zweck der Datenverarbeitung ist also die Warnung Betroffener. Die Nutzbarkeit ist an die Rolle Online-Service gebunden. Diese Rolle wird jedem Online-Dienst im Verbund für die Dauer des Bestehens des Verbundes zugewiesen. Die entsprechende `Util`-Politik ist im Anhang als Listung 2 zu finden.

Umsetzung: Überblick über das Verfahren

Die Online-Dienste, die Betroffene von Leaks unter ihren Nutzern warnen möchten, bilden einen Verbund. Dieser kann zeitlich begrenzt sein. Die teilnehmenden Online-Dienste müssen einander nicht notwendigerweise bekannt sein. Die Dienste erhalten Kenntnis der Leaks über einen weiteren Informationskanal. Dies kann z.B. das Abonnieren eines Leakage-Informationsdienstes sein[8]. Innerhalb dieses Verbunds fragt jeder Dienst vor dem Warnen eines Betroffenen beim Warnzähler-Dienst an, ob der Betroffene bereits von einem anderen Dienst im Verbund gewarnt wurde. Ist dies der Fall, verzichtet der anfragende Dienst auf ein erneutes Warnen. Andernfalls warnt er den Betroffenen. Dieser kann daraufhin ein Ändern des Passworts veranlassen.
Die technische Umsetzung des Verfahrens wird möglich, indem durch Vereinbarung eines geheimen gemeinsamen Schlüssels ein Verbund hergestellt wird. Damit die Services nicht voneinander erfahren, kommunizieren sie zur Vereinbarung des Schlüssels über einen Mediator. Das detaillierte Verfahren ist in den Protokollen 1 und 2 der bereits veröffentlichten Arbeit [91] zu finden. Die Sicherheit des Verfahrens basiert auf der Sicherheit des erweiterten Diffie-Hellman-Schlüsselaustauschs [142] und der Annahme, dass der Mediator dem Angreifermodell Honest-but-Curious [122] folgt. Dementsprechend agiert er protokollkonform, nutzt aber möglicherweise gesammelte Information, um Klartextdaten zu inferieren. Werden keine weiteren Schutzmaßnahmen ergriffen, sind ihm zum Beispiel die Identitäten der Online-Dienste bekannt.
Erhält ein Online-Dienst Kenntnis von der Betroffenheit eines Nutzers von einem Leak, so fragt er den Warnzähler-Dienst an, ob der Betroffene bereits von einem anderen Dienst im Verbund gewarnt wurde. Damit weder Warnzähler-Dienst noch andere Dienste die Identität des Nutzers erfahren, pseudonymisiert der Online-Dienst das anzufragende personenbezogene Datum maßgeschneidert für die Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Elementrelation. Hierzu nutzt er das in Algorithmus 9 gelistete Verfahren. Dies entspricht den Schritten 1 und 2 von Protokoll 3 der Arbeit [91][9].
Um nun die Nutzbarkeit auswerten zu können, sendet der Online-Dienst die Bloom-Filter-Kodierung des Pseudonyms an den Mediator. Dies entspricht Schritt 3 von Protokoll 3 der Arbeit [91][10]. Der anfragende Dienst entspricht hierbei dem Analysten des Auswertungsverfahrens nach Algorithmus 10 in Kapitel 5 dieser Arbeit.
Der Mediator nimmt die Rolle des in Algorithmus 10 zur Auswertung der Nutzbarkeit definierten Dekryptors ein. Zuvor wurde von ihm in der Rolle des Schlüsselmanagers bzw. Pseudonymisierers bereits den Bloom-Filter mit denselben Parametern generiert, die auch bei der Schlüsselvereinbarung an die Online-Services ausgerollt wurden. Der Bloom-Filter ist initial leer. Im Laufe des

[8] siehe z.B. der Leak-Checker der Universität Bonn `https://leakchecker.uni-bonn.de`.
[9] bzw. Protokoll 25 im Anhang.
[10] bzw. Protokoll 25 im Anhang.

Bestehens des Verbunds wird er mit den angefragten Kodierungen gefüllt. Er entspricht stets der nutzbarkeitserhaltend pseudonymisierten Menge der bereits gewarnten Nutzer der im Verbund kollaborierenden Online-Dienste. Auf dieser Menge führt er nun die Schritte 4, 5 und 6 des Protokolls 3 aus [91][11] durch. Dies entspricht der Auswertung der Nutzbarkeit durch den Dekryptor und die Rückmeldung des Ergebnisses an den Analysten im Auswertungsverfahren 10 aus Kapitel 5.

Fazit zum Leakage-Warning-Management

Mit den vorgestellten Protokollen wurde eine Möglichkeit aufgezeigt, Warnungen Betroffener bei gleichzeitiger Wahrung ihrer Privatsphäre zu verwalten. Im Gegensatz zu Diensten, die die Daten im Klartext austauschen, wird hierbei eine wesentliche Verbesserung des Vertraulichkeitsschutzes erreicht. Die Nutzung eines Bloom-Filters stärkt diesen Ansatz zusätzlich durch den Umstand, dass die Pseudonymisierung nicht mehr aus voneinander differenzierbaren Pseudonymen besteht. Vielmehr können sich die Kodierungen der Pseudonyme im Bloom-Filter überlappen. Dies erwirkt eine zusätzliche Reduzierung der beim Warndienst vorliegenden Information. Somit kann der Warndienst ohne erfolgte konkrete Abfrage nicht mit Sicherheit bestimmen, ob eine aus dem Bloom-Filter ermittelte Kodierung tatsächlich einer Kodierung eines Pseudonyms oder einer aus Überlappungen von Kodierungen entstandenen Kombination von Einsen entspricht. Ein weiterer Vorteil ist die Reduzierung des Speicherplatzes [91].

7.3 Fazit

Im vorliegenden Kapitel werden zwei Anwendungsbeispiele aus unterschiedlichen Anwendungsszenarien beschrieben. So wird anhand exemplarischer Beispiele die Sinnhaftigkeit der Anforderungsformulierung gemäß Kapitel 3 und 4, der Pseudonymisierungsverfahren aus Kapitel 5 und der in Kapitel 6 entsprechend formulierten Übersetzungsregeln veranschaulicht. Diese entsprechen den Komponenten des in Kapitel 2 der vorliegenden Dissertation vorgestellten Rahmenwerks.
Es bleibt festzuhalten, dass die Komponenten des Rahmenwerks in den verschiedenen Anwendungen in unterschiedlichem Umfang zum Einsatz kommen. Im Beispiel der Umfrage-Plattform kann die Formulierbarkeit einer Kombination einer Vielzahl von Nutzbarkeitsanforderungen erfordern. Im Beispiel eines Online-Dienstes, der einzelne personenbezogene Daten auf ein Vorkommen in einer ihm ansonsten unbekannten Datenmenge testen möchte, wird hingegen lediglich eine sehr eingeschränkte Anzahl von Nutzbarkeitsanforderungen benötigt.
Insgesamt kann festgestellt werden, dass das Rahmenwerk für die Pseudonymisierung personenbezogener Daten in unterschiedlichen Anwendungsfällen und Anforderungen flexibel nutzbar ist. Verglichen mit dem Fehlen eines solchen Rahmenwerks als Hilfsmittel wird eine deutliche Verbesserung der Situation von Anwendern von Pseudonymisierung erreicht. Weiterhin kann festgehalten werden, dass das Rahmenwerk ausgehend von vom Anwender formulierten Nutzbarkeitsanforderungen automatisiert maßgeschneiderte Pseudonymisierungen erstellt. Hierbei muss der Anwender keine Kenntnis der einzusetzenden Privacy-Enhancing-Technologies vorweisen. Wo erforderlich,

[11] bzw. Protokoll 25 im Anhang

bietet `Util` durch geforderte Angaben von Zweck der Datenverarbeitung und Ablaufdatum der Speicherungsdauer die erforderliche Hilfestellung. Darüber hinaus bleibt die Umsetzung von Vertraulichkeitsanforderungen dem Anwender gegenüber transparent. Zur weiteren Unterstützung des Anwenders bei der Formulierung von `Util`-Politiken können Politik-Editoren mit grafischen Nutzeroberflächen entwickelt werden. Ein Beispiel eines solchen Editors für die Adressierung insbesondere von Algorithmen des maschinellen Lernens wurde im Rahmen der Forschungsarbeiten dieser Dissertation in einer studentischen Abschlussarbeit betreut [148]. Abbildung 5 im Anhang dieser Arbeit zeigt das in der studentischen Arbeit erarbeitete Beispiel.

8 Zusammenfassung, Fazit und Ausblick

Im vorliegenden Kapitel werden zunächst in Abschnitt 8.1 die Ergebnisse dieser Dissertation zusammengefasst. In welchem Umfang die in Abschnitt 1.1 eingeführten Forschungsfragen durch die Ergebnisse beantwortet wurden, wird in Abschnitt 8.2 beantwortet. Schließlich werden in Abschnitt 8.3 Forschungsfragen und -ideen dargestellt, die ausgehend von den Ergebnissen dieser Dissertation untersucht werden sollten.

8.1 Zusammenfassung

Anwender verschiedener Disziplinen verarbeiten personenbezogene Daten. Um das Risiko der Verletzung der Privatsphäre Betroffener und des unintendierten Abfließens von Information aus personenbezogenen Daten zu mindern, sollen die Daten pseudonymisiert werden. Die Anwender stehen vor der Herausforderung, Pseudonymisierungsverfahren so anzuwenden, dass der Schutz der Vertraulichkeit sinnvoll gegeben ist, gleichzeitig aber die Nutzbarkeit der Daten für den Anwendungsfall erhalten bleibt. Bisher sind in der Literatur keine Ansätze bekannt, die dies flexibel nach dem Stand der Wissenschaft für Anwender ermöglichen.
In dieser Dissertation wurde in Kapitel 2 ein Rahmenwerk entworfen und in den Kapiteln 3, 4, 5 und 6 ausgearbeitet, das die nutzbarkeitserhaltende Datenpseudonymisierung zum Schutz der Vertraulichkeit personenbezogener Daten bei deren Verarbeitung erleichtert. Damit wurde eine Möglichkeit bereitgestellt, die Anwendern unterschiedlicher Disziplinen einen einheitlichen Zugang zu Datenpseudonymisierung systematisch nach dem Stand der Wissenschaft ermöglicht. Gleichzeitig zeigt das Rahmenwerk auf, dass eine nutzbarkeitserhaltende Datenpseudonymisierung durch Orientierung an den in einem Anwendungsfall erforderlichen Nutzbarkeiten möglich ist.
Für das Rahmenwerk wurden zunächst in Kapitel 3 Klassen von Anforderungen an die Nutzbarkeit und die Vertraulichkeit der Daten erarbeitet. Hierbei wurden mehrere Designentscheidungen getroffen, beschrieben und motiviert. Eine Folge war, dass die Anforderungsklassen der Nutzbarkeiten bereits unter Berücksichtigung der Vertraulichkeitsanforderungen entworfen wurden. Dadurch wurde die spätere Übersetzung in entsprechende Pseudonymisierungsverfahren weiter vereinheitlicht und erleichtert. Weiterhin weisen die Klassen die Möglichkeit der Identifizierung und Ergänzung weiterer Anforderungen auf. Ein Beispiel ist das Hinzufügen weiterer Relationen zur Anforderungsklasse der Verkettbarkeit bezüglich einer bestimmten Relation.
Für eine erleichterte automatisierte Formulierung der Anforderungen an Daten wurde in Kapitel 4 `Util`, eine Sprache zur maschinen- und menschenlesbaren Anforderungsformulierung erarbeitet. Auch hier wurden Designentscheidungen getroffen und umgesetzt. Diese führten dazu, dass die Sprache die Nutzbarkeitsanforderungen an Teilmengen einer von einer Nutzbarkeitspolitik

adressierten Datensammlung orientiert auflistet. Eine weitere Folge ist, dass der für die Berücksichtigung von Vertraulichkeitsanforderungen vom Anwender zu betreibende Aufwand reduziert wurde. Durch das gezielte Kombinieren von attribut- und knotenbasierten Formulierungen der Anforderungskomponenten konnte ein Grad an Übersichtlichkeit hergestellt werden, der die Menschenlesbarkeit erleichtert.
Die in Kapitel 5 vorgestellten Pseudonymisierungsverfahren wurden entsprechend dem Stand der Wissenschaft in den PET und der Angewandten Kryptographie ermittelt. Jedoch wurde bei der Auswahl der Verfahren deren Praktikabilität berücksichtigt. Ein Beispiel für die Auswirkungen dieser Berücksichtigung ist die Wahl partiell homomorpher statt vollhomomorpher Verschlüsselungsverfahren. Dies geht zwar mit Einschränkungen bei der Umsetzbarkeit von zu berechnenden Funktionen auf den Pseudonymen einher. Jedoch kann im Vergleich zum Einsatz vollhomomorpher Verfahren eine im Ressourcenbedarf praktikablere Umsetzung erfolgen. Ein weiteres Beispiel ist die Wahl eines im Rahmen der Betreuung von Abschlussarbeiten entstandenen Verfahrens für das k-Means-Clustering als Beispiel für ein Pseudonymisierungsverfahren für die Erhaltung der Nutzbarkeit der Anforderung Algorithmus der Ausprägung k-Means. Im Gegensatz zu Verfahren aus der Literatur [110] kommt dieses Verfahren ohne die Aufteilung der Klartextdaten und die Durchführung der Datenpseudonymisierung bei verschiedenen Parteien aus. Jedoch ist in dem in dieser Arbeit angeführten Verfahren der Zugriff auf eine vertrauenswürdige Entität wie z.B. ein Trusted-Execution-Environment erforderlich.
In Kapitel 6 wurde die Verbindung zwischen der Anforderungsformulierung und der Anforderungsumsetzung hergestellt. Hierfür wurden Regeln zur Übersetzung der entsprechenden `Util`-Elemente in geeignet parametrisierte Pseudonymisierungsverfahren erarbeitet.
Die Verwendung des Rahmenwerks wurde in Kapitel 7 an zwei unterschiedlichen Anwendungsbeispielen exemplarisch demonstriert. Hierdurch konnte dargestellt werden, dass das Rahmenwerk innerhalb verschiedener Einsatzszenarien für die nutzbarkeitserhaltende Datenpseudonymisierung eingesetzt werden kann.

8.2 Fazit

In diesem Abschnitt wird zusammenfassend dargestellt, wie die Beantwortung der Forschungsfragen durch die vorliegende Dissertation erreicht wurde. Ziel war die Erarbeitung eines Rahmenwerks, mit dem Pseudonymisierungen ohne PET-Expertenkenntnisse das Risiko der Reidentifizierung Betroffener reduzierend und ganz bestimmte Nutzbarkeiten erhaltend erstellt und verarbeitet werden können. Aus diesem Ziel wurden die in Kapitel 1.1 formulierten vier Forschungsfragen abgeleitet. Diese wurden in der Dissertation beantwortet.
Die **erste Forschungsfrage** befasst sich damit, wie Anforderungen an Pseudonymisierungen, die aus der intendierten Nutzung dieser stammen, vor der Erstellung der Pseudonymisierung formuliert werden können. Zur Beantwortung dieser Frage wurde in Kapitel 3 eine Systematisierung von Anforderungsklassen vorgestellt. Diese sind grundsätzlich in Vertraulichkeits- und Nutzbarkeitsanforderungen unterteilt. Diese Unterteilung der Anforderungsklassen spiegelt den Gegensatz der Schutzziele Vertraulichkeit und Verfügbarkeit von Information wieder. Die Gegensätze wurden aus-

balanciert und für eine Privatsphäre respektierende Verarbeitung der Daten in Einklang gebracht. Die Anforderungsklassen erlauben die Formulierung von Nutzbarkeitsanforderungen, die eine Vielzahl geplanter Datenverarbeitungen ermöglichen. Gleichzeitig bieten die Vertraulichkeitsanforderungen die Möglichkeit, Anforderungen zur Limitierung der Verfügbarkeit der Nutzbarkeiten by-Design und bereits vor Erstellung der Pseudonymisierung zu formulieren.
Die **zweite Forschungsfrage** behandelt die Erstellung von für die Nutzbarkeitsanforderungen maßgeschneiderten Pseudonymisierungen. Zur Beantwortung der Frage wurden in Kapitel 5 für jede der erarbeiteten Anforderungsklassen exemplarisch für mindestens eine Nutzbarkeitsanforderung Pseudonymisierungsverfahren vorgestellt, die den einzelnen Anforderungen genügen. Bei der Auswahl der zugrundeliegenden PET aus der wissenschaftlichen Literatur wurde neben der Umsetzung der Nutzbarkeit der Vertraulichkeitsschutz und die Praktikabilität in Form von mit existierenden kryptographischen Verfahren vergleichbarem Ressourcenaufwand berücksichtigt.
Die **dritte Forschungsfrage** handelt von der Überlegung, wie formulierte Nutzbarkeitsanforderungen zur automatisierten Erstellung von maßgeschneiderten Pseudonymisierungen verwendet werden können. Zur Beantwortung dieser Frage wurde mit `Util` in Kapitel 4 eine `XML`-basierte Beschreibungssprache für die maschinen- und menschenlesbare Formulierung von Nutzbarkeitsanforderungen vorgestellt. In Kapitel 6 wurde eine Datenstruktur zur Verarbeitung automatisiert erstellter Pseudonymisierungen und Übersetzungsregeln zur Ableitung maßgeschneiderter, geeignet parametrisierter Pseudonymisierungen aus den Anforderungen vorgestellt.
Die maschinen- und menschenlesbare Formulierung von Nutzbarkeitsanforderungen umfasst zum einen die Definierung von Strukturen und Elementen für die Formulierung von Nutzbarkeitsanforderungen. Zum anderen werden die Vertraulichkeitsanforderungen berücksichtigt. Hierbei wird dem Umstand Rechnung getragen, dass die anzunehmende Anwenderin nicht zwingend Expertenwissen auf dem Gebiet der PET aufweist. Um eine Anforderungsformulierung mit einem möglichst geringen Wissensumfang zu erleichtern, wurden implizite und explizite Anforderungen unterschieden. Implizite Vertraulichkeitsanforderungen wurden bereits beim Entwurf des Rahmenwerks berücksichtigt. Entsprechend sind sie in das Design der Anforderungsklassen, der Auswahl der Pseudonymisierungsverfahren, der Übersetzungsregeln von `Util`-Elementen in die Pseudonymisierung und der Umsetzungsstruktur der Pseudonymisierung eingeflossen. Lediglich Vertraulichkeitsanforderungen, die durch die Angabe bestimmter Parameter eine aktive Formulierung durch den Anwender erfordern, wurden als explizite Anforderungen mit entsprechenden Elementen in `Util` berücksichtigt.
Schließlich bietet eine Nutzbarkeitspolitik eine übersichtliche Darstellung der von einer Pseudonymisierung erfüllten Anforderungen. Diese dient der Dokumentierung der Verarbeitung personenbezogener Daten.
Die **vierte und letzte Forschungsfrage** behandelt die strukturelle Umsetzung einer automatisiert erstellten, für Nutzbarkeitsanforderungen maßgeschneiderte Pseudonymisierung unter Berücksichtigung von Vertraulichkeitsanforderungen. Zur Beantwortung dieser letzten Frage wurde in Kapitel 6 eine Datenstruktur vorgestellt, in der für eine Datensammlung Pseudonyme für verschiedene Nutzbarkeiten einheitlich verwaltet und adressiert werden können. Zusätzlich werden Datenschutzprinzipien berücksichtigt. Diese Berücksichtigung hatte die Entscheidung zur Folge, Pseudonyme und zusätzliche Daten, die unmittelbar oder mittelbar zur Aufdeckung der zugrunde-

liegenden Klartextdaten geeignet sind, getrennt zu speichern. Dies soll eine Steuerung des Zugriffs auf die in Pseudonymisierungen enthaltene Information und eine Ergänzung mit weiteren Sicherheitsmechanismen erleichtern. Ein Beispiel ist hier die Steuerung des Zugriffs auf Schlüssel durch Zugriffskontroll- und Schlüsselmanagementsysteme.

Mit dem Rahmenwerk und seinen Komponenten

- **Anforderungsmodell** mit den Anforderungsklassen Nutzbarkeitsanforderungen und Vertraulichkeitsanforderungen
- **Anforderungsbeschreibung** mit der Beschreibungssprache für Nutzbarkeitspolitiken `Util`
- **Übersetzungsregeln** von Nutzbarkeitspolitiken in parametrisierte nutzbarkeitserhaltende Pseudonymisierungsverfahren
- **Anforderungsumsetzung** als Pseudonymisierung mit Utility-Tag-Struktur und Secrets-Struktur

wurde ein Beitrag zur Verbreitung des Einsatzes risikomindernder, am Stand der Wissenschaft orientierter, explizit bestimmte Nutzbarkeiten erhaltender Pseudonymisierung durch Anwender geleistet.

Die dauerhafte Speicherung und Verarbeitung personenbezogener Daten im Klartext geht häufig mit dem Risiko des Zugriffs unautorisierter Dritter einher. Analog zu dem in [149] skizzierten Lösungsvorschlag sollen Anwender ohne PET-Expertenwissen in die Lage versetzt werden, Pseudonymisierung effektiv zu nutzen. Insgesamt soll durch die Ergebnisse der Dissertation mittel- bis langfristig der Umfang der Nutzung von personenbezogenen Klartextdaten reduziert und durch risikomindernde Datenaufbereitungen ersetzt werden. Dass die Umsetzung in unterschiedlichen Anwendungen bereits möglich ist, wurde durch die Anwendungsbeispiele in Kapitel 3 nachgewiesen. Insbesondere für die Verarbeitung sensibler Daten wie im Banken- oder medizinischen Bereich scheint dies in Hinblick auf immer stärkere zur Verfügung stehende Ressourcen zur Datenverarbeitung realisierbar[1].

8.3 Ausblick

In der vorliegenden Dissertation wurden Komponenten für ein Rahmenwerk entwickelt, das eine automatisierte nutzbarkeitserhaltende Pseudonymisierung personenbezogener Daten ermöglicht. In den Abschnitten 8.1 und 8.2 wurde dargelegt, wie dieses Ziel mit dem erarbeiteten Rahmenwerk erreicht wurde. Über die Forschungsfragen hinaus wurden während der Forschungsarbeiten und der Erstellung der Dissertation weitere, von der Autorin als forschungsrelevant erachtete Fragestellungen identifiziert. Diese werden im Folgenden skizziert.

[1] Analog zum Mooreschen Gesetz [111] wird angenommen, dass der auf Rechnern verfügbare Speicherplatz und die Rechenkapazität weiter ansteigen wird.

Optimierung der Datenstruktur und der Pseudonymisierungsverfahren Die in Kapitel 6 vorgestellte Datenstruktur erlaubt trotz der Unterschiede in der Struktur und Umsetzung der einzelnen Pseudonymisierungsverfahren eine einheitliche Adressierung, Darstellung und Umsetzung von nutzbarkeitserhaltenden Pseudonymisierung. Dies ermöglicht eine automatisierte Umsetzung von nutzbarkeitserhaltender Pseudonymisierung als Alternative zur Klartextverarbeitung. Hier können durch weitere Designbetrachtungen Optimierungen des Speicherplatzbedarfs erreicht werden. Eine noch zu erforschende Frage ist zum Beispiel, wie die mehrfache Verwendung von Schlüsseln zur Reduzierung der replizierten Speicherung dieser bei gleichzeitigem Erhalt der Automatisierbarkeit und der Flexibilität der Anwendung realisiert werden kann.

Anonymisierung mit Fallunterscheidung Für einige Nutzbarkeiten existieren Anonymisierungsverfahren, die ohne die Erfordernis der Umsetzung von Zusatzmechanismen wie z.B. Monitoring im verarbeitenden System auskommen. Dennoch bieten diese Verfahren eine möglicherweise größere Minderung des Restrisikos der Reidentifizierung Betroffener. Ein Beispiel ist ein sogenanntes Cloaking-Verfahren wie das in [41] vorgestellte. Es kann als Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Elementrelation betrachtet werden. Vergleichsmenge ist hierbei die Menge aller innerhalb einer Cloaking-Region vorkommenden Standorte. Ist die Aufdeckung der Standorte nicht erforderlich, so könnte mittels des beschriebenen Verfahrens k-Anonymität auf Standortdaten umgesetzt werden. Verallgemeinert ist zunächst zu ermitteln, für welche Kombinationen von Nutzbarkeitsanforderungen Daten nicht nur pseudonymisiert, sondern hinreichend nutzbarkeitserhaltend anonymisiert werden können. Hierbei muss eruiert werden, wie weit Einbußen bei der Genauigkeit der Berechnungsergebnisse tolerierbar sind. Eine Erweiterung des Rahmenwerks um eine entsprechende, auf solche Kombinationen als Fallunterscheidung prüfende Komponente würde die automatisierte Umsetzung von Anonymisierungsverfahren bereits pseudonymisierter Daten ermöglichen. Ein Beispiel ist das Anonymisieren bereits nutzbarkeitserhaltend pseudonymisierter Daten mittels Differentially-Private-Verfahren [51, 153].

Weitere Unterstützung des Anwenders Das Rahmenwerk kann den Anwender bei der Formulierung der Politik zusätzlich unterstützen, indem es eine Benutzeroberfläche mit für Anwendungsfälle vordefinierten Politiken bereitstellt. Hierzu ist zu erarbeiten, welche Anforderungen innerhalb eines Anwendungsfalls besonders häufig in Kombination miteinander vorkommen. Im Vergleich zur mit `Util` bereitgestellten Nutzerschnittstelle würde eine solche Oberfläche die Formulierung von Anforderungen weiter erleichtern. Die Bedienbarkeit der zu entwickelnden Benutzeroberfläche kann durch Ergebnisse von Nutzerstudien weiter verbessert werden.

Seitenkanäle und Risikoabschätzung, Härtung der Pseudonymisierung Eine Grundannahme der vorliegenden Arbeit ist, dass personenbezogene Daten durch die Informationsreduktion der Pseudonymisierung in eine Ebene der syntaktischen Informationsrepräsentation überführt werden, die die Verfügbarkeit verschiedener Information aus den Klartextdaten zumindest verringert. In Kapitel 5 dieser Arbeit wurde für einzelne Pseudonymisierungsverfahren herausgearbeitet, wie durch die Auswahl der zugrundeliegenden PET weiterer, über die Nutzbarkeit hinausgehender un-

intendierter Abfluss von Information gemindert werden kann. Hier ist zu untersuchen, wie sich die kombinierte Bereitstellung von Pseudonymen unterschiedlicher Nutzbarkeiten für dieselben Daten auf den zusätzlichen, unintendierten Informationsabfluss auswirkt. Erste interne Untersuchungen zeigen, dass insbesondere die Verkettbarkeit bezüglich Gleichheit auf verschiedenen Datenattributen und stark eingegrenzter Klartextdatenmenge eine Inferierung der Klartextdaten erleichtert. Weitere Kombinationen und entsprechende Maßnahmen zur Härtung sind zu erarbeiten.
Eine `Util`-Politik dient auch als Dokumentation der durch die Pseudonymisierung verfügbaren Nutzbarkeiten. Es könnte erarbeitet werden, inwiefern das bloße Wissen aus der Politik für Reidentifizierungsangriffe genutzt werden kann. Eine weitere Frage ist, wie eine Politik zur Risikoabschätzung der Reidentifizierung Betroffener genutzt werden kann. Insgesamt könnten die Ansätze von Durak et al. [50], Neumann et al. [116, 127] und Kohlmayer et al. [94] miteinander in Einklang gebracht, auf das Rahmenwerk übertragen und das Ergebnis analysiert werden. Analog zu Modellen und Algorithmen in der physikalischen Kryptoanalyse[2] können Modelle zur Entwicklung systematischer Reidentifizierungsangriffe auf Pseudonymisierungen entwickelt werden. Diese könnten zur besseren Einschätzung des Grad des Vertraulichkeitsschutzes von Pseudonymisierungsverfahren verwendet werden.

Mehrschichtige Sicherheit In der vorliegenden Arbeit wurde dem Schutzziel der Vertraulichkeit personenbezogener Daten durch die Erleichterung von Datenpseudonymisierung begegnet. Weitere Fragestellungen zum Schutzziel der Integrität der Daten könnten davon ausgehend in der Zukunft behandelt werden. Im Sinne von mehrschichtiger Sicherheit [80] sind verschiedene Schutzziele durch die Kombinierung unterschiedlicher technisch-organisatorischer Maßnahmen zu untersuchen und umzusetzen. Beispiele sind die rollenbasierte Zugriffskontrolle [57] und die Einschränkung der Verfügbarkeit von Berechnungsergebnissen durch das Monitoring der Verarbeitung pseudonymisierter Daten als technisch-organisatorische Schutzmaßnahme. Insbesondere bei der Umsetzung des Rahmenwerks in der Praxis sind hier weitere Vorbereitungen erforderlich.

Berücksichtigung des Standes der Technik Schätzungen zufolge sinkt die Halbwertzeit der Technik und des Wissens in der IT-Sicherheit [157, 126]. Es ist zu erwarten, dass vollhomomorphe Verschlüsselung und Verfahren zum effektiven Monitoring der Datenverarbeitung und der Durchsetzung der Begrenzung dieser in den kommenden Jahren stark an Effektivität gewinnen werden. Bereits heute wird untersucht, wie den Auswirkungen des Einsatzes von Quantencomputern auf die Sicherheit und den Datenschutz begegnet werden kann. Insbesondere bei der Umsetzung des Rahmenwerks müssen diese Auswirkungen und die Möglichkeit der Auswechselbarkeit veralteter und dadurch nicht mehr sicherer Pseudonymisierungsverfahren mittelfristig berücksichtigt werden. Bei der Langzeitspeicherung von Daten, aber auch bei der zukünftigen Angreifbarkeit von in Datenlecks gefundenen verschlüsselten Daten muss dies besonders berücksichtigt werden.

Alternative, branchenspezifische Anforderungsklassen Das in der Arbeit herausgearbeitete Rahmenwerk wurde unabhängig von einem konkreten Anwendungsfall entworfen. Wie in Kapitel

[2] Siehe hierzu z.B. die Arbeit von Lemke-Rust [100].

7 aufgezeigt, erlauben dadurch insbesondere die Anforderungsklassen eine Beschreibung von Nutzbarkeiten aus unterschiedlichen Anwendungsfällen. Dies hat den Vorteil, dass Wissen um die nutzbarkeitserhaltende Datenpseudonymisierung nach dem Stand der Wissenschaft unterschiedlichen Anwendungsgebieten zugänglich gemacht werden kann. Somit soll verhindert werden, dass jede Branche, in der personenbezogene Daten pseudonymisiert verarbeitet werden sollen, einander stark ähnelnde Verfahren oder sogar schwache Verfahren neu entwickelt. Es stellt sich jedoch die Frage, ob durch eine spezialisiertere, für bestimmte Branchen angepasste Formulierung von Anforderungen eine weitere Erleichterung der Nutzbarkeit des Rahmenwerks für den Anwender erreicht werden kann.

Abschließend kann festgehalten werden, dass das Forschungsgebiet der nutzbarkeitserhaltenden Datenpseudonymisierung und der PET eine Vielzahl offener Forschungsfragen bereithält. In dieser Dissertation wurde der Stand der Wissenschaft um ein Rahmenwerk erweitert, das die maßgeschneidert nutzbarkeitserhaltende, für den Anwender umsetzbare, die Vertraulichkeit personenbezogener Daten schützende und somit das Risiko der Reidentifizierung Betroffener mindernde Datenpseudonymisierung ermöglicht.

Literaturverzeichnis

[1] Acar, Abbas u. a.: „A survey on homomorphic encryption schemes: Theory and implementation“. In: *ACM Computing Surveys (CSUR)* 51.4 (2018), S. 1–35.

[2] *ADAC Fahr + Spar Telematik-Versicherung*. URL: `https://www.adac.de/produkte/versicherungen/autoversicherung/fahr-und-spar` (besucht am 27. 11. 2021).

[3] Agrawal, Rakesh u. a.: „Order preserving encryption for numeric data“. In: *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. 2004, S. 563–574.

[4] Agrawal, Rakesh u. a.: „XPref: a preference language for P3P“. In: *Computer Networks* 48.5 (2005), S. 809–827.

[5] Lauradoux, Cedric et al.: *Data Pseudonymisation: Advanced Techniques and Use Cases: Technical analysis of cybersecurity measures in data protection and privacy*. Techn. Ber. European Union Agency for Cybersecurity (ENISA), 2021.

[6] Andreea, Ionita: „Private Set Intersection: Past Present and Future“. In: *Proceedings of the 18th International Conference on Security and Cryptography-SECRYPT*. 2021, S. 680–685.

[7] Arp, Daniel u. a.: „Privacy-Enhanced Fraud Detection with Bloom Filters“. In: *International Conference on Security and Privacy in Communication Systems*. Springer. 2018, S. 396–415.

[8] Ashley, Paul u. a.: „Enterprise privacy authorization language (EPAL)“. In: *IBM Research* 30 (2003), S. 31.

[9] Azraoui, Monir u. a.: „A-PPL: an accountability policy language“. In: *Data privacy management, autonomous spontaneous security, and security assurance*. Springer, 2014, S. 319–326.

[10] Barbaro, Michael ; Zeller, Tom ; Hansell, Saul: „A face is exposed for AOL searcher no. 4417749“. In: *New York Times* 9.2008 (2006), S. 8.

[11] Bauer, Friedrich L.: „Cryptosystem“. In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg ; Sushil Jajodia. Boston, MA: Springer US, 2011, S. 284–285. URL: `https://doi.org/10.1007/978-1-4419-5906-5_167`.

[12] Becker, Moritz Y ; Fournet, Cédric ; Gordon, Andrew D: *SecPAL: Design and semantics of a decentralized authorization language*. Techn. Ber. Technical Report MSR-TR-2006-120, Microsoft Research, 2006.

[13] Becker, Moritz Y ; Fournet, Cédric ; Gordon, Andrew D: „SecPAL: Design and semantics of a decentralized authorization language“. In: *Journal of Computer Security* 18.4 (2010), S. 619–665.

[14] Becker, Moritz Y ; Malkis, Alexander ; Bussard, Laurent: „A framework for privacy preferences and data-handling policies“. In: *Microsoft Research Cambridge Technical Report, MSR-TR-2009-128* (2009).

[15] Bellare, Mihir ; Rogaway, Phillip: „Introduction to modern cryptography“. In: *Ucsd Cse* 207 (2005), S. 207.

[16] Bellare, Mihir ; Rogaway, Phillip: „Optimal asymmetric encryption“. In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer. 1994, S. 92–111.

[17] Biryukov, Alex: „Chosen Plaintext Attack“. In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg ; Sushil Jajodia. Boston, MA: Springer US, 2011, S. 205–206. url: https://doi.org/10.1007/978-1-4419-5906-5_557.

[18] Biryukov, Alex: „Dictionary Attack (I)“. In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg ; Sushil Jajodia. Boston, MA: Springer US, 2011, S. 332–332. url: https://doi.org/10.1007/978-1-4419-5906-5_571.

[19] Biryukov, Alex ; Khovratovich, Dmitry: *Related-key Cryptanalysis of the Full AES-192 and AES-256*. Cryptology ePrint Archive, Report 2009/317. https://ia.cr/2009/317. 2009.

[20] Bishop Christopher, M u. a.: „Pattern recognition and machine learning“. In: *Information science and statisticsNew York: Springer* (2006).

[21] Biskup, Joachim ; Flegel, Ulrich: „Transaction-Based Pseudonyms in Audit Data for Privacy Respecting Intrusion Detection“. In: 2000, S. 28–48.

[22] Bissmeyer, Norbert ; Petit, Jonathan ; Bayarou, Kpatcha M: „CoPRA: Conditional pseudonym resolution algorithm in VANETs“. In: *2013 10th annual conference on wireless on-demand network systems and services (WONS)*. IEEE. 2013, S. 9–16.

[23] Bloom, Burton H: „Space/time trade-offs in hash coding with allowable errors“. In: *Communications of the ACM* 13.7 (1970), S. 422–426.

[24] Bogatov, Dmytro ; Kollios, George ; Reyzin, Leonid: „A comparative evaluation of order-revealing encryption schemes and secure range-query protocols“. In: *Proceedings of the VLDB Endowment* 12.8 (2019), S. 933–947.

[25] Boneh, Dan u. a.: „Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation“. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2015, S. 563–594.

[26] Bonnoron, Guillaume u. a.: „Somewhat/fully homomorphic encryption: Implementation progresses and challenges“. In: *International Conference on Codes, Cryptology, and Information Security*. Springer. 2017, S. 68–82.

[27] Borking, John: „Der identity protector“. In: *Datenschutz und Datensicherheit* 20.11 (1996), S. 654–658.

[28] Borking, John J ; Raab, Charles: „Laws, PETs and other technologies for privacy protection“. In: *Journal of Information, Law and Technology* 1 (2001), S. 1–14.

[29] Bost, Raphael u. a.: „Machine Learning Classification over Encrypted Data“. In: *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2014*. 2015. URL: http://www.internetsociety.org/doc/machine-learning-classification-over-encrypted-data.

[30] Brakerski, Zvika ; Gentry, Craig ; Vaikuntanathan, Vinod: „(Leveled) Fully Homomorphic Encryption Without Bootstrapping“. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS '12. Cambridge, Massachusetts: ACM, 2012, S. 309–325. URL: http://doi.acm.org/10.1145/2090236.2090262.

[31] Brekne, Tønnes ; Årnes, André ; Øslebø, Arne: „Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies“. In: *International Workshop on Privacy Enhancing Technologies*. Springer. 2005, S. 179–196.

[32] BSI: „Kryptographische Verfahren: Empfehlungen und Schlüssellangen“. In: *Technische Richtlinie TR-02102-1, Bundesamt fur Sicherheit in der Informationstechnik* (2017).

[33] BSI: „Kryptographische Verfahren: Empfehlungen und Schlüssellangen, Version 2021-01“. In: *Technische Richtlinie TR-02102-1, Bundesamt fur Sicherheit in der Informationstechnik* (2021).

[34] *Bundesdatenschutzgesetz*. 2017. URL: https://www.gesetze-im-internet.de/bdsg_2018/ (besucht am 25. 01. 2022).

[35] Chang, Shu-jen u. a.: „Third-round report of the SHA-3 cryptographic hash algorithm competition“. In: *NIST Interagency Report* 7896 (2012), S. 121.

[36] *Charta der Grundrechte der Europäischen Union*. 2010. URL: https://www.europarl.europa.eu/germany/resource/static/files/europa_grundrechtecharta/_30.03.2010.pdf.

[37] Chen, Hao ; Laine, Kim ; Player, Rachel: „Simple encrypted arithmetic library-SEAL v2.1“. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2017, S. 3–18.

[38] Chenette, Nathan u. a.: „Practical order-revealing encryption with limited leakage“. In: *International conference on fast software encryption*. Springer. 2016, S. 474–493.

[39] Chenette, Nathan et al: „Practical Order-Revealing Encryption with Limited Leakage“. In: *Fast Software Encryption*. Hrsg. von Thomas Peyrin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, S. 474–493.

[40] Chor, Benny u. a.: „Private information retrieval“. In: *Proceedings of IEEE 36th Annual Foundations of Computer Science*. IEEE. 1995, S. 41–50.

[41] Christin, Delphine u. a.: „A distributed privacy-preserving mechanism for mobile urban sensing applications“. In: *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2015, S. 1–6.

[42] Costan, Victor ; Devadas, Srinivas: „Intel sgx explained.“ In: *IACR Cryptol. ePrint Arch.* 2016.86 (2016), S. 1–118.

[43] Couper, Mick P u. a.: „Risk of disclosure, perceptions of risk, and concerns about privacy and confidentiality as factors in survey participation“. In: *Journal of official statistics* 24.2 (2008), S. 255.

[44] Cranor, Lorrie Faith: „P3P: Making privacy policies more useful“. In: *IEEE Security & Privacy* 1.6 (2003), S. 50–55.

[45] Daemen, Joan ; Rijmen, Vincent: „AES proposal: Rijndael“. In: (1999).

[46] *Datenschutz-Wiki*. 2021. url: https://www.datenschutz-wiki.de.

[47] Delaune, Stéphanie ; Jacquemard, Florent: „Decision Procedures for the Security of Protocols with Probabilistic Encryption against Offline Dictionary Attacks“. In: *Journal of Automated Reasoning* 36 (2006).

[48] Doganay, Mahir Can u. a.: „Distributed privacy preserving k-means clustering with additive secret sharing“. In: *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*. 2008, S. 3–11.

[49] Dolev, Danny ; Dwork, Cynthia ; Naor, Moni: „Nonmalleable cryptography“. In: *SIAM review* 45.4 (2003), S. 727–784.

[50] Durak, F. Betül ; DuBuisson, Thomas M. ; Cash, David: „What Else is Revealed by Order-Revealing Encryption?“ In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, S. 1155–1166. url: https://doi.org/10.1145/2976749.2978379.

[51] Dwork, Cynthia: „Differential privacy: A survey of results“. In: *International conference on theory and applications of models of computation*. Springer. 2008, S. 1–19.

[52] Dworkin, Morris J: *SHA-3 standard: Permutation-based hash and extendable-output functions*. Techn. Ber. 2015.

[53] Eckert, Claudia: *IT-Sicherheit: Konzepte-Verfahren-Protokolle*. Walter de Gruyter, 2013.

[54] El Gamal, Taher: „A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms“. In: *Proceedings of CRYPTO 84 on Advances in Cryptology*. Santa Barbara, California, USA: Springer-Verlag New York, Inc., 1985, S. 10–18. url: http://dl.acm.org/citation.cfm?id=19478.19480.

[55] „Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)“. In: *Official Journal of the European Union* L119/59 (2016).

[56] Fan, Jinliang u. a.: „Prefix-Preserving IP Address Anonymization: Measurement-Based Security Evaluation and a New Cryptography-Based Scheme“. In: *Comput. Netw.* 46.2 (2004), S. 253–272. url: https://doi.org/10.1016/j.comnet.2004.03.033.

[57] Ferraiolo, DF ; Kuhn, DR: „Role-Based Access Control“. In: *Proceedings of 15th National Computer Security Conference*. 1992.

[58] Field, Andy ; Hole, Graham: *How to design and report experiments*. Sage, 2002.

[59] Flegel, Ulrich: „Praktikabler Datenschutz für Log-Daten“. In: *Proceedings of the 10th DFN-CERT Workshop on Sicherheit in vernetzten Systemen, DFN-CERT publications, pages F1–F20, Hamburg, Germany*. 2003.

[60] Flegel, Ulrich: „Pseudonymizing Unix log files“. In: *International Conference on Infrastructure Security*. Springer. 2002, S. 162–179.

[61] Flegel, Ulrich ; Biskup, Joachim: „Requirements of information reductions for cooperating intrusion detection agents“. In: *International Conference on Emerging Trends in Information and Communication Security*. Springer. 2006, S. 466–480.

[62] Flegel, Ulrich ; Hoffmann, Johannes ; Meier, Michael: „Cooperation enablement for centralistic early warning systems“. In: *Proceedings of the 2010 ACM Symposium on Applied Computing*. 2010, S. 2001–2008.

[63] Flegel, Ulrich ; Meier, Michael: „Authorization architectures for privacy-respecting surveillance“. In: *European Public Key Infrastructure Workshop*. Springer. 2007, S. 1–17.

[64] Freedman, Michael J ; Nissim, Kobbi ; Pinkas, Benny: „Efficient private matching and set intersection“. In: *International conference on the theory and applications of cryptographic techniques*. Springer. 2004, S. 1–19.

[65] Gentry, Craig u. a.: „Fully homomorphic encryption using ideal lattices.“ In: *STOC*. Bd. 9. 2009, S. 169–178.

[66] Gerl, Armin: „Modelling of a privacy language and efficient policy-based de-identification“. Diss. Université de Lyon und Universität Passau (Deutschland), 2019.

[67] Gerl, Armin ; Bölz, Felix: „Layered Privacy Language Pseudonymization Extension for Health Care.“ In: *Studies in health technology and informatics* 264 (2019), S. 1189–1193.

[68] Gerl, Armin ; Prey, Florian: „LPL Personal privacy policy user interface: design and evaluation“. In: *Mensch und Computer 2018-Workshopband* (2018).

[69] *Gesetze im Internet*. url: `https://www.gesetze-im-internet.de` (besucht am 10. 04. 2022).

[70] Ghafir, Ibrahim u. a.: „A survey on network security monitoring systems“. In: *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. IEEE. 2016, S. 77–82.

[71] Goldreich, Oded: *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[72] GroupLens: *MovieLens*. 2020. url: `https://grouplens.org/datasets/movielens/`.

[73] Gunasinghe, Hasini ; Bertino, Elisa: „RahasNym: Pseudonymous identity management system for protecting against linkability“. In: *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. IEEE. 2016, S. 74–85.

[74] Hansen, Marit ; Jensen, Meiko ; Rost, Martin: „Protection goals for privacy engineering“. In: *2015 IEEE Security and Privacy Workshops*. IEEE. 2015, S. 159–166.

[75] Heen, Olivier ; Neumann, Christoph: „On the privacy impacts of publicly leaked password databases“. In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer. 2017, S. 347–365.

[76] Heurix, Johannes u. a.: „LiDSec- A Lightweight Pseudonymization Approach for Privacy-Preserving Publishing of Textual Personal Information“. In: *2012 Seventh International Conference on Availability, Reliability and Security* 00 (2011), S. 603–608.

[77] Hofstetter, Max: „Scalable Privacy-Preserving Study Platform“. Masterarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn, 2019.

[78] SurveyMonkey Inc.: *SurveyMonkey*. url: https://www.surveymonkey.com (besucht am 01.01.2021).

[79] Iyilade, Johnson ; Vassileva, Julita: „P2U: A Privacy Policy Specification Language for Secondary Data Sharing and Usage“. In: Bd. 2014. 2014, S. 18–22.

[80] Jajodia, Sushil ; Kogan, Boris: *Integrating an object-oriented data model with multilevel security*. Techn. Ber. DAYTON UNIV OH, 1990.

[81] Jajodia, Sushil ; Tilborg, Henk CA van van: *Encyclopedia of Cryptography and Security*. Springer, 2011.

[82] Jensen, Meiko ; Lauradoux, Cedric ; Limniotis, Konstantinos: *Pseudonymisation techniques and best practices: Recommendations on shaping technology according to data protection and privacy provisions*. Techn. Ber. European Union Agency for Cybersecurity (ENISA), 2021.

[83] Jha, Somesh ; Kruger, Luis ; McDaniel, Patrick: „Privacy preserving clustering“. In: *European symposium on research in computer security*. Springer. 2005, S. 397–417.

[84] Jonas, Stephan ; Siewert, Simon ; Spreckelsen, Cord: „Privacy-Preserving Record Grouping and Consent Management Based on a Public-Private Key Signature Scheme: Theoretical Analysis and Feasibility Study“. In: *J Med Internet Res* (). url: http://www.ncbi.nlm.nih.gov/pubmed/30977738.

[85] Al-Kadit, Ibrahim A: „Origins of cryptology: The Arab contributions“. In: *Cryptologia* 16.2 (1992), S. 97–126.

[86] Kaliski, Burt: „Asymmetric Cryptosystem“. In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg ; Sushil Jajodia. Boston, MA: Springer US, 2011, S. 49–50. url: https://doi.org/10.1007/978-1-4419-5906-5_394.

[87] Kasem-Madani, Saffija: „A framework for encrypted computation on shared data“. In: *Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit*. Hrsg. von Michael Meier ; Delphine Reinhardt ; Steffen Wendzel. Bonn: Gesellschaft für Informatik e.V., 2016, S. 191–196.

[88] Kasem-Madani, Saffija ; Meier, Michael: *Security and Privacy Policy Languages: A Survey, Categorization and Gap Identification*. 2015. arXiv: 1512.00201 [cs.CR].

[89] Kasem-Madani, Saffija ; Meier, Michael: „Utility Requirement Description for Utility-Preserving and Privacy-Respecting Data Pseudonymization“. In: *International Conference on Trust and Privacy in Digital Business*. Springer. 2020, S. 171–185.

[90] Kasem-Madani, Saffija ; Meier, Michael ; Wehner, Martin: „Towards a Toolkit for Utility and Privacy-Preserving Transformation of Semi-structured Data Using Data Pseudonymization“. In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Hrsg. von Joaquin Garcia-Alfaro u. a. Cham: Springer International Publishing, 2017, S. 163–179.

[91] Kasem-Madani, Saffija u. a.: „Privacy-Preserving Warning Management for an Identity Leakage Warning Network“. In: *EICC 2020: European Interdisciplinary Cybersecurity Conference, Rennes, France, November 18, 2020*. ACM, 2020, 4:1–4:6. URL: https://doi.org/10.1145/3424954.3424955.

[92] Khandelwal, Ankesh u. a.: „Analyzing the air language: a semantic web (production) rule language“. In: *International Conference on Web Reasoning and Rule Systems*. Springer. 2010, S. 58–72.

[93] Kipnis, Aviad ; Hibshoosh, Eliphaz: „Efficient Methods for Practical Fully Homomorphic Symmetric-key Encrypton, Randomization and Verification.“ In: *IACR Cryptol. ePrint Arch.* 2012 (2012), S. 637.

[94] Kohlmayer, Florian ; Lautenschläger, Ronald ; Prasser, Fabian: „Pseudonymization for research data collection: is the juice worth the squeeze?“ In: *BMC medical informatics and decision making* 19.1 (2019), S. 1–7.

[95] Krämer, Markus: „*k*-Means auf Chiffraten“. Bachelorarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn, 2017.

[96] Krämer, Markus: „Privacy Preserving Clustering in the Pseudonymisation Tool“. Masterarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn, 2020.

[97] Kumaraguru, Ponnurangam ; Calo, Seraphin: „A survey of privacy policy languages“. In: *Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security, ACM*. 2007.

[98] *Las Vegas Man Sentenced To Prison For Credit Card Fraud Scheme*. 2021. URL: https://www.justice.gov/usao-nv/pr/las-vegas-man-sentenced-prison-credit-card-fraud-scheme.

[99] Leicht, Jens ; Heisel, Maritta: „A Survey on Privacy Policy Languages: Expressiveness Concerning Data Protection Regulations“. In: *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*. 2019, S. 1–6.

[100] Lemke-Rust, Kerstin: *Models and algorithms for physical cryptanalysis*. Dissertation, Ruhr-Universität Bochum, 2007.

[101] Lloyd, Stuart: „Least squares quantization in PCM“. In: *IEEE transactions on information theory* 28.2 (1982), S. 129–137.

[102] Madarie, Renushka u. a.: „Stolen account credentials: an empirical comparison of online dissemination on different platforms“. In: *Journal of Crime and Justice* 42.5 (2019), S. 551–568.

[103] Malderle, Timo: „Bedrohung durch Identitätsdatendiebstahl“. In: (2021).

[104] McCurley, Kevin S: „The discrete logarithm problem“. In: *Proc. of Symp. in Applied Math*. Bd. 42. USA. 1990, S. 49–74.

[105] Meints, Martin: „Implementierung großer biometrischer Systeme“. In: *Datenschutz und Datensicherheit-DuD* 31.3 (2007), S. 189–193.

[106] Meskine, Fatima ; Bahloul, Safia Nait: „Privacy preserving k-means clustering: a survey research.“ In: *Int. Arab J. Inf. Technol.* 9.2 (2012), S. 194–200.

[107] Meyer, Daniel: „Ausgewählte Angriffe auf Pseudonymisierungen mit Verfügbarkeitsoptionen“. Bachelorarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn, 2017.

[108] Meyer, Daniel: „Utility-driven pseudonymization for a flexible survey platform“. Masterarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn, 2020.

[109] Mohan, Arun Prasad ; Gladston, Angelin u. a.: „Merkle tree and Blockchain-based cloud data auditing“. In: *International Journal of Cloud Applications and Computing (IJCAC)* 10.3 (2020), S. 54–66.

[110] Mohassel, Payman ; Rosulek, Mike ; Trieu, Ni: „Practical Privacy-Preserving K-means Clustering“. In: *Proceedings on Privacy Enhancing Technologies* 2020 (2019), S. 414–433.

[111] Moore, Gordon E u. a.: *Cramming more components onto integrated circuits*. 1965.

[112] Morel, Victor ; Pardo, Raúl: „SoK: Three Facets of Privacy Policies“. In: *Proceedings of the 19th Workshop on Privacy in the Electronic Society*. WPES'20. Virtual Event, USA: Association for Computing Machinery, 2020, S. 41–56. url: https://doi.org/10.1145/3411497.3420216.

[113] Motwani, Rajeev ; Xu, Ying: „Efficient algorithms for masking and finding quasi-identifiers“. In: *Proceedings of the Conference on Very Large Data Bases (VLDB)*. 2007, S. 83–93.

[114] Neubauer, Thomas ; Heurix, Johannes: „A methodology for the pseudonymization of medical data“. In: *International journal of medical informatics* 80.3 (2011), S. 190–204.

[115] Neubauer, Thomas ; Riedl, Bernhard: „Improving patients privacy with Pseudonymization“. In: *Studies in health technology and informatics* 136 (2008), S. 691.

[116] Neumann, Geoffrey K u. a.: „Pseudonymization risk analysis in distributed systems“. In: *Journal of Internet Services and Applications* 10.1 (2019), S. 1–16.

[117] Oliveira, Stanley R. M. ; Zaiane, Osmar R: „Privacy Preserving Clustering by Data Transformation“. In: *J. Inf. Data Manag.* 1 (2003), S. 37–52.

[118] Paar, Christof ; Pelzl, Jan: *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[119] Paillier, Pascal: „Public-Key Cryptosystems Based on Composite Degree Residuosity Classes“. In: *Advances in Cryptology, EUROCRYPT 99*. Hrsg. von Jacques Stern. Bd. 1592. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, S. 223–238.

[120] Pallas, Frank ; Grambow, Martin: „Three tales of disillusion: Benchmarking property preserving encryption schemes“. In: *International Conference on Trust and Privacy in Digital Business*. Springer. 2018, S. 39–54.

[121] Parsian, Mahmoud: *Data Algorithms: Recipes for Scaling Up with Hadoop and Spark*. 1st. O'Reilly Media, Inc., 2015.

[122] Paverd, AJ ; Martin, Andrew ; Brown, Ian: *Modelling and Automatically Analysing Privacy Properties for Honest-but-Curious Adversaries*. Techn. Ber.

[123] Popa, Raluca Ada u. a.: „CryptDB: Protecting Confidentiality with Encrypted Query Processing“. In: *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. SOSP '11. Cascais, Portugal: ACM, 2011, S. 85–100. url: http://doi.acm.org/10.1145/2043556.2043566.

[124] Postel, J.: *Internet Protocol*. RFC 791 (INTERNET STANDARD). Updated by RFCs 1349, 2474, 6864. Internet Engineering Task Force, 1981. URL: http://www.ietf.org/rfc/rfc791.txt.

[125] Cambridge University Press, Hrsg.: *Cambridge Dictionary*. 2014.

[126] Rashotte, Rob: „Closing the cyber skills gap requires a culture of continuous learning". In: *Security Magazine* (2020). URL: https://www.securitymagazine.com/articles/94254-closing-the-cyber-skills-gap-requires-a-culture-of-continuous-learning (besucht am 11.01.2022).

[127] Rocher, Luc ; Hendrickx, Julien M ; De Montjoye, Yves-Alexandre: „Estimating the success of re-identifications in incomplete datasets using generative models". In: *Nature communications* 10.1 (2019), S. 1–9.

[128] Rossow, Olaf: *Datenschutz-Notizen*. 2021. URL: https://www.datenschutz-notizen.de/pseudonym-oder-anonym-das-unbekannte-wesen-in-der-dsgvo-4127941/.

[129] Rost, Martin ; Pfitzmann, Andreas: „Datenschutz-Schutzziele revisited". In: *Datenschutz und Datensicherheit-DuD* (2009).

[130] Sako, Kazue: „Semantic Security". In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg ; Sushil Jajodia. Boston, MA: Springer US, 2011, S. 1176–1177. URL: https://doi.org/10.1007/978-1-4419-5906-5_23.

[131] Samarati, Pierangela ; Sweeney, Latanya: „Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression". In: (1998).

[132] Scaiano, Martin u. a.: „A unified framework for evaluating the risk of re-identification of text de-identification tools". In: *Journal of biomedical informatics* 63 (2016), S. 174–183.

[133] Schaad, Andreas u. a.: „Optimized and controlled provisioning of encrypted outsourced data". In: *Proceedings of the 19th ACM symposium on Access control models and technologies*. 2014, S. 141–152.

[134] Schools, W3: *XML Tutorial*. URL: https://www.w3schools.com/xml/default.asp (besucht am 06.01.2022).

[135] Schulze Darup, Moritz: „Encrypted polynomial control based on tailored two-party computation". In: *International Journal of Robust and Nonlinear Control* 30.11 (2020), S. 4168–4187.

[136] Schwartmann, Rolf ; Weiss, Steffen (Editors): *White Paper on Pseudonymization Drafted by the Data Protection Focus Group for the Safety, Protection, and Trust Platform for Society and Businesses in Connection with the 2017 Digital Summit*. Techn. Ber. Digital Summit's data protection focus group, 2017.

[137] *SELinux Project*. URL: https://www.nsa.gov/what-we-do/research/selinux/ (besucht am 01.01.2022).

[138] Shafranovich, Yakov: *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. RFC 4180. 2005. URL: https://rfc-editor.org/rfc/rfc4180.txt.

[139] Shamir, Adi: „How to Share a Secret“. In: *Commun. ACM* 22.11 (1979), S. 612–613. URL: http://doi.acm.org/10.1145/359168.359176.

[140] Slagell, Adam J ; Lakkaraju, Kiran ; Luo, Katherine: „FLAIM: A Multi-level Anonymization Framework for Computer and Network Logs.“ In: *LISA*. Bd. 6. 2006, S. 3–8.

[141] Standard, OASIS: „extensible access control markup language (xacml) version 3.0“. In: *A:(22 January 2013). URl: http://docs. oasis-open. org/xacml/3.0/xacml-3.0-core-spec-os-en. html* (2013).

[142] Steiner, Michael ; Tsudik, Gene ; Waidner, Michael: „Diffie-Hellman key distribution extended to group communication“. In: *Proceeding of the 3rd ACM conference on Computer and Communications Security*. 1996, S. 31–37.

[143] *The FreeBSD Project*. URL: www.freebsd.org/de/ (besucht am 01. 01. 2022).

[144] Thiele, Clemens: „BGH: Dynamische IP-Adressen als personenbezogene Daten“. In: *Zeitschrift für Informationsrecht* 2017.4 (2017), S. 411–418.

[145] Trabelsi, Slim ; Sendor, Jakub ; Reinicke, Stefanie: „PPL: PrimeLife Privacy Policy Engine“. In: *POLICY 2011, IEEE International Symposium on Policies for Distributed Systems and Networks, Pisa, Italy, 6-8 June 2011*. IEEE Computer Society, 2011, S. 184–185. URL: https://doi.org/10.1109/POLICY.2011.24.

[146] Vimercati, Sabrina de Capitani di ; Foresti, Sara: „Quasi-Identifier“. In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg ; Sushil Jajodia. Boston, MA: Springer US, 2011, S. 1010–1011. URL: https://doi.org/10.1007/978-1-4419-5906-5_763.

[147] Waage, Tim ; Wiese, Lena: „Property preserving encryption in NoSQL wide column stores“. In: *OTM Confederated International ConferencesÖn the Move to Meaningful Internet Systems"*. Springer. 2017, S. 3–21.

[148] Wehner, Martin: „Utility-driven Requirement Description of Machine Learning Techniques“. Masterarbeit. Rheinische Friedrich-Wilhelms-Universität Bonn, 2018.

[149] Wendzel, Steffen ; Kasem-Madani, Saffija: „IoT Security: The Improvement-Decelerating 'Cycle of Blame'“. In: *Journal of Cyber Security and Mobility* (2016).

[150] Wendzel, Steffen u. a.: „Pattern-Based Survey and Categorization of Network Covert Channel Techniques“. In: *ACM Comput. Surv.* 47.3 (2015). URL: https://doi.org/10.1145/2684195.

[151] Working Party 29, Directorate C (Fundamental Rights ; Union Citizenship), European Commiss: *Opinion 05/2014 on Anonymisation Techniques*. 2014.

[152] Xiao, Liangliang ; Bastani, Osbert ; Yen, I-Ling: „An Efficient Homomorphic Encryption Protocol for Multi-User Systems.“ In: *IACR Cryptol. EPrint Arch.* 2012 (2012), S. 193.

[153] Xiong, Ping ; Zhu, Tian-Qing ; Wang, Xiao-Feng: „A survey on differential privacy and applications“. In: *Jisuanji Xuebao/Chinese Journal of Computers* 37.1 (2014), S. 101–122.

[154] Yagisawa, Masahiro: „Fully homomorphic encryption without bootstrapping.“ In: *IACR Cryptol. EPrint Arch.* 2015 (2015), S. 474.

[155] Yang, Jean ; Yessenov, Kuat ; Solar-Lezama, Armando: „A Language for Automatically Enforcing Privacy Policies“. In: *SIGPLAN Not.* 47.1 (2012), S. 85–96. URL: https://doi.org/10.1145/2103621.2103669.

[156] Zanga, Adrian: „Using ElectionGuard for secure remote voting on untrusted devices“. In: (2020).

[157] Zehnder, Carl August: „Der Informatikberuf ist keine Zirkusnummer“. In: *Computerworld* (2007). URL: https://www.computerworld.ch/business/digitalisierung/informatikberuf-zirkusnummer-1444200.html (besucht am 11.01.2022).

[158] Zhang, Xuyun u. a.: „An efficient quasi-identifier index based approach for privacy preservation over incremental data sets on cloud“. In: *Journal of Computer and System Sciences* 79.5 (2013), S. 542–555. URL: https://www.sciencedirect.com/science/article/pii/S0022000012001766.

[159] Zimmer, Ephraim u. a.: „PEEPLL: privacy-enhanced event pseudonymisation with limited linkability“. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 2020, S. 1308–1311.

Liste der Algorithmen

Abbildungsverzeichnis

Tabellenverzeichnis

f

Selbstständigkeitserklärung

Hiermit versichere ich, die vorliegende Dissertation ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Bonn, 24. Januar 2023

Saffija Kasem-Madani

Anhang

Beispiele für eine Instanziierung der Policy-Sprache `Util`

Im Folgenden werden für die in Kapitel 7 beschriebenen Anwendungsbeispiele `Util`-Formulierungen für geeignete Nutzbarkeitspolitiken gelistet. Diese müssen vom Nutzer angegeben werden.

`Util`-Politik für das Anwendungsbeispiel Umfrageplattform

Die gelistete Nutzbarkeitspolitik entspricht auf Umfragedaten formulierte Nutzbarkeitsanforderungen einer Umfrageplattform, wie sie in Kapitel 7.1 beschrieben ist. Da die Formulierung für alle Datenattribute dieselbe ist, wurde die Darstellung der Politik auf das Datenattribut `age` gekürzt. Zur Veranschaulichung wurden die Nutzbarkeitsanforderungen Verkettbarkeit bezüglich der Gleichheit, Verkettbarkeit bezüglich der Relation Kleiner-Gleich, Addition und Multiplikation formuliert. Zur Veranschaulichkeit expliziter Vertraulichkeitsanforderung wurde an die Nutzbarkeitsanforderung Aufdeckbarkeit die Bedingung als `Condition` formuliert, dass das Ergebnis der Durchführung der Addition auf derselben Datenmenge kleiner oder gleich dem Wert 0.1 ist.

```
1
2  <?xml version="2.0"?>
3
4  <UtilityPolicy>
5    <Data type="set" id=54321>
6
7
8        <!-------Nutzbarkeitsanforderung Verkettbarkeit bzgl. Gleichheit auf allen
       Einträgen des Datenattributs Age ----->
9        <DataAttributeSet id=12345>
10         <DataAttribute>
11           <Name>Age</Name>
12             <Range>
13               <!-- Die Set id der Klartextdatenmenge zur Referenzierung; muss eine
       Teilmenge der in DataAttributeSet referenzierten id sein.  -->
14               <Set id=12345>
15                 <AllDataEntries />
16               </Set>
17             </Range>
18         </DataAttribute>
19         <Utility>
20           <Requirement type="Linkability" id=0>
21             <Relation>
22               equality
23             </Relation>
24             <On>
```

```
25              <Pseudonyms Position = 0>
26                <SetBySetID>
27                  12345
28                </SetBySetID>
29              </Pseudonyms>
30              <Pseudonyms Position = 1>
31                <SetBySetID>
32                  12345
33                </SetBySetID>
34              </Pseudonyms>
35            </On>
36          </Requirement>
37          <Bindings>
38            <Binding type="Purpose">
39              Analyse A
40            </Binding>
41          </Bindings>
42        </Utility>
43      </DataAttributeSet>
44      <!-------Nutzbarkeitsanforderung Verkettbarkeit bzgl. Kleiner-Gleich auf allen
        Einträgen des Datenattributs Age ----->
45      <DataAttributeSet id=12345>
46        <DataAttribute>
47          <Name>Age</Name>
48          <Range>
49            <!-- Die Set id der Klartextdatenmenge zur Referenzierung; muss eine
        Teilmenge der in DataAttributeSet referenzierten id sein.  -->
50            <Set id=12345>
51              <AllDataEntries />
52            </Set>
53          </Range>
54        </DataAttribute>
55        <Utility>
56          <Requirement type="Linkability" id=1>
57            <Relation>
58              less-equal
59            </Relation>
60            <On>
61              <Pseudonyms Position = 0>
62                <SetBySetID>
63                  12345
64                </SetBySetID>
```

```
65              </Pseudonyms>
66              <Pseudonyms Position = 1>
67                <SetBySetID>
68                  12345
69                </SetBySetID>
70              </Pseudonyms>
71            </On>
72          </Requirement>
73          <Bindings>
74            <Binding type="Purpose">
75              Analyse A
76            </Binding>
77          </Bindings>
78        </Utility>
79      </DataAttributeSet>
80
81      <!-------Nutzbarkeitsanforderung Operation Addition auf allen Einträgen des
        Datenattributs Age ----->
82      <DataAttributeSet id=12345>
83        <DataAttribute>
84          <Name>Age</Name>
85          <Range>
86            <!-- Die Set id der Klartextdatenmenge zur Referenzierung; muss eine
        Teilmenge der in DataAttributeSet referenzierten id sein.  -->
87            <Set id=12345>
88              <AllDataEntries />
89            </Set>
90          </Range>
91        </DataAttribute>
92        <Utility>
93          <Requirement type=‘‘Operation’’ id=2>
94            <Type>
95              Addition
96            </Type>
97          </Requirement>
98          <Bindings>
99            <Binding type="Purpose">
100             Analyse A
101           </Binding>
102         </Bindings>
103       </Utility>
104     </DataAttributeSet>
```

```
105
106
107       <!-------Nutzbarkeitsanforderung Operation Multiplikation auf allen Einträgen
          des Datenattributs Age ----->
108       <DataAttributeSet id=12345>
109         <DataAttribute>
110           <Name>Age</Name>
111           <Range>
112             <!-- Die Set id der Klartextdatenmenge zur Referenzierung; muss eine
          Teilmenge der in DataAttributeSet referenzierten id sein.  -->
113             <Set id=12345>
114               <AllDataEntries />
115             </Set>
116           </Range>
117         </DataAttribute>
118         <Utility>
119           <Requirement type=‘‘Operation’’ id=3>
120             <Type>
121               Multiplication
122             </Type>
123           </Requirement>
124           <Bindings>
125             <Binding type="Purpose">
126               Analyse A
127             </Binding>
128           </Bindings>
129         </Utility>
130       </DataAttributeSet>
131
132
133       <!-- Nutzbarkeitsanforderung Aufdeckbarkeit mit Condition -->
134       <DataAttributeSet id=12345>
135         <DataAttribute>
136           <Name>Age</Name>
137           <Range>
138             <!-- Die Set id der Klartextdatenmenge zur Referenzierung; muss eine
          Teilmenge der in DataAttributeSet referenzierten id sein.  -->
139             <Set id=12345>
140               <AllDataEntries />
141             </Set>
142           </Range>
143         </DataAttribute>
```

```
144         <Utility>
145           <Requirement type=‘‘Disclosability’’  id=4/>
146             <Conditions>
147               <Condition type = opResult>
148                 <OperationID>
149                   2
150                 </OperationID>
151                 <SetID>
152                   12345
153                 </SetID>
154                 <comparanceRelation>
155                   less-equal
156                 </comparanceRelation>
157                 <comparanceValue>
158                   0.1
159                 </comparanceValue>
160               </Condition>
161             </Conditions>
162             <Bindings>
163               <Binding type="Purpose">
164                 Analyse A
165               </Binding>
166             </Bindings>
167         </Utility>
168       </DataAttributeSet>
169     </Data>
170   </UtilityPolicy>
```

Listing 1: *Formulierung einer Nutzbarkeitspolitik in Util für eine Analyse A ein Datenattribut Age mit Nutzbarkeitsanforderungen Verkettbarkeit bezüglich der zweistelligen Gleichheitsrelation, der zweistelligen Kleiner-Gleich-Relation und der Aufdeckbarkeit.*

Util-Politik für das Anwendungsbeispiel Privacy-preserving Leakage-Warning-Management

Die in diesem Abschnitt gelistete Nutzbarkeitspolitik entspricht Nutzbarkeitsanforderungen, wie sie in Kapitel 7.2 beschrieben sind. In diesem Beispiel handelt es sich bei den geleakten Daten um E-Mail-Adressen.

```
1
2  xml version="2.0"?>
3  tilityPolicy>
4  <Data type="set" id=54321>
5    <!-------Nutzbarkeitsanforderung Verkettbarkeit bzgl. Gleichheit auf allen Einträ
       gen des Datenattributs email ----->
6    <DataAttributeSet id=54321>
7      <DataAttribute>
8        <Name>email</Name>
9        <Range>
10         <!-- Die Set id der Klartextdatenmenge zur Referenzierung; muss eine
       Teilmenge der in DataAttributeSet referenzierten id sein.  -->
11         <Set id=54321>
12           <AllDataEntries />
13         </Set>
14       </Range>
15     </DataAttribute>
16     <Utility>
17       <Requirement type="Linkability" id=0>
18         <Relation>
19           element-of
20         </Relation>
21         <On>
22           <Pseudonyms Position = 0>
23             <SetBySetID>
24               <!-- Menge der Klartextdaten, die abgefragt werden darf. Bezieht sich
       auf die dem Mediator als Bloomfilter vorliegende Menge -->
25               5678
26             </SetBySetID>
27           </Pseudonyms>
28           <Pseudonyms Position = 1>
29             <SetBySetID>
30               <!-- Diese ID ist verknüpft mit der durch den aktuellen Schlü
       sselaustausch begründeten Verbund -->
31               externalL
32             </SetBySetID>
```

```
33            </Pseudonyms>
34          </On>
35        </Requirement>
36        <Bindings>
37          <Binding type="Purpose">
38            Leak-Warnung
39          </Binding>
40          <Binding type="Role">
41          Online-Service
42          </Binding>
43        </Bindings>
44      </Utility>
45    </DataAttributeSet>
46  </Data>
47  UtilityPolicy>
```

Listing 2: *Formulierung einer Nutzbarkeitspolitik in Util für die Nutzbarkeitsanforderung Verkettbarkeit bezüglich der zweistelligen Elementrelation mit einer Menge von geleakten E-Mail-Adressen von bereits Gewarnten.*

Anwendungsbeispiel Privacy-preserving Leakage-Warning-Management: Protokolle

Im Folgenden werden die Protokolle aus [91] gelistet. Hierbei handelt es sich um eine Ergänzung der Beschreibung der Anwendung in Kapitel 7.2 und eine Umsetzung der in Kapitel 3.2 beschriebenen Nutzbarkeitsanforderung der Verkettbarkeit bezüglich der Elementrelation. Analog zu den in Kapitel 1.2.1 eingeführten Rollen nimmt der Mediator $\mathcal{M}$ in den Protokollen 23 und 24 die Rolle des Schlüsselmanagers ein. Die zu pseudonymisierenden Daten D liegen den Online-Diensten $\mathcal{OD}_1, \cdots, \mathcal{OD}_n$ im Klartext vor. Sie nehmen die Rolle des Pseudonymisierers ein. Schließlich nimmt der Mediator in Protokoll 25 die Rolle des Analysten ein.

Laufzeit und Grössen der Pseudonymisierung für die Nutzbarkeitsanforderung Verkettbarkeit bezüglich der Elementrelation nach [91]

Für einen in Python implementierten Bloomfilter wurde bei 50000 E-Mail-Adressen im Klartext eine deutliche Verringerungen der Speichergröße im Verlgeich zur Speicherung der Klartextdaten beobachtet. Dies gilt auch für 100000 und 150000 Klartexte und ist in Abbildung 2 ersichtlich.
Die Laufzeit einer Abfrage nach Protokoll 25 steigt in Abhängigkeit der Füllrate leicht an. Dies kann in Abbildung 3 nachvollzogen werden.

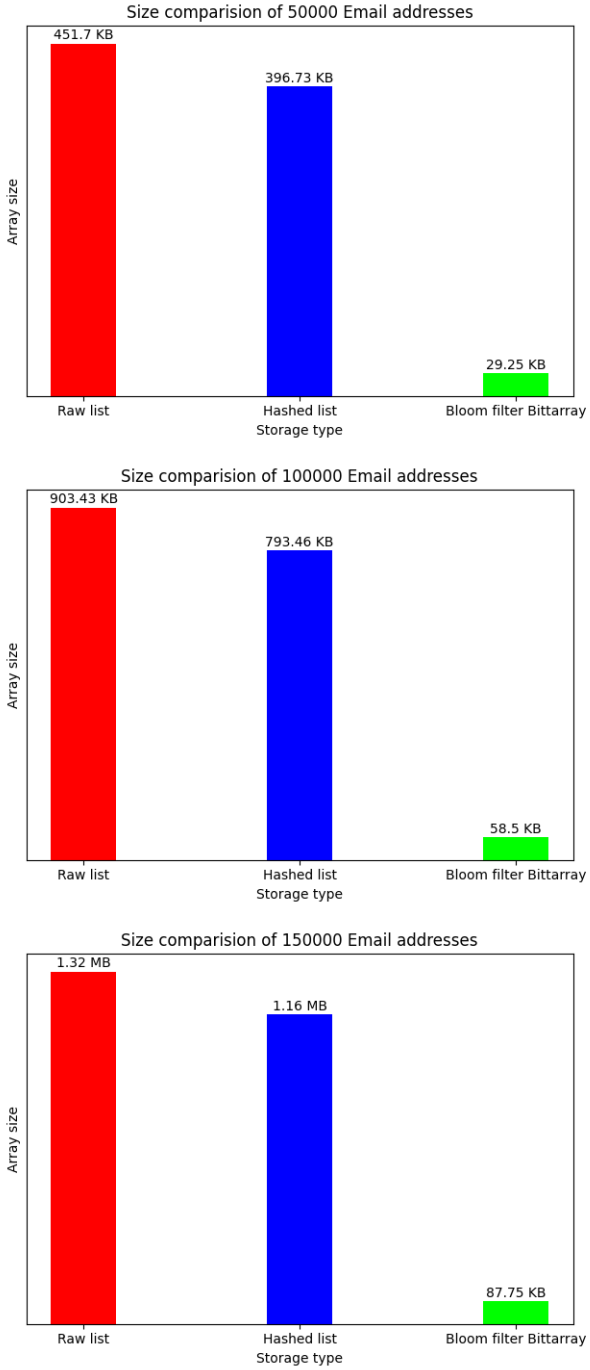

Abbildung 2: *Vergleich des Speicherplatzes zwischen E-Mail-Adressen im Klartext (rot), mit Salt gehasht (blau) und pseudonymisiert (grün).*

Eingabe : Online-Dienste $\mathcal{OD}_1, \cdots, \mathcal{OD}_n$, die ihre Nutzer warnen wollen, falls diese als Betroffene eines Daten-Leaks erkannt werden.
Ausgabe : Online-Dienste $\mathcal{OD} \in \{\mathcal{OD}_1, \cdots, \mathcal{OD}_n\}$ tauschen über den Mediator $\mathcal{M}$ öffentliche Diffie-Hellman- und Bloom-Filter-Parameter untereinander aus. Über diese Parameter hinaus erfahren die Online-Dienste nichts übereinander.

1 $\mathcal{M}$ erzeugt zufällig Primzahlen p und g angemessener Länge als öffentliche Diffie-Hellman-Parameter, wobei $\frac{p-1}{2}$ prim ist, und einen initial leeren Bloom-Filter $\mathcal{BF}$;
2 $\mathcal{M}$ erzeugt g, p und die Konfiguration des Bloom-Filters $\mathcal{BF}$;
3 **for** $\mathcal{OD}_i$, $1 \leq i \leq n-1$ **do**
4 | $\mathcal{OD}_i$ erzeugt einen geheimen Diffie-Hellman-Parameter $sc_i \in Z_p^+$;
5 | $\mathcal{M}$ sendet g, p und die Konfiguration des Bloom-Filters $\mathcal{BF}$ an $\mathcal{OD}_i$;
6 **end**

Algorithmus 23 : Protokoll 1 aus [91]: Vorbereitung und Austausch der öffentlichen Parameter and die teilnehmenden Online-Dienste.

Eingabe : Online-Dienste $\mathcal{OD}_1, \cdots, \mathcal{OD}_n$ mit Diffie-Hellman-Parametern p, g.
Ausgabe : Die Online-Dienste vereinbaren über den Mediator $\mathcal{M}$ einen gemeinsamen geheimen Schlüssel κ. Der Mediator selbst lernt den Schlüssel nicht.

1 **for** $i = 1, \ldots, n-1$ **do**
2 | $\mathcal{OD}_i$ erzeugt ein zufälliges $sc_i \in Z_p^*$;
3 | $\mathcal{OD}_i$ berechnet die Zwischenwerte $S_i := \{g^{sc_1 \cdots i / sc_j} | j \in \{1, \cdots, i\}\}$ und $g^{sc_1 \cdots sc_i}$ und sendet diese an $\mathcal{M}$;
4 | $\mathcal{M}$ sendet die Zwischenwerte S_i und $g^{sc_1 \cdots sc_i}$ an $\mathcal{OD}_{i+1}$;
5 **end**
6 ;
7 $\mathcal{OD}_n$ erzeugt ein zufälliges $sc_n \in Z_p^*$.;
8 $\mathcal{OD}_n$ berechnet die Zwischenwerte $S_n := \{g^{sc_1 \cdots sc_n / sc_i} | i \in \{1, \cdots, n\}\}$ und sendet diese an $\mathcal{M}$;
9 $\mathcal{M}$ sendet S_n an alle $\mathcal{OD}_i$;
10 **for** $i = 1, \ldots, n$ **do**
11 | $\mathcal{OD}_i$ berechnet den gemeinsamen geheimen Schlüssel $\kappa = g^{sc_1 \cdots sc_n}$ unter Nutzung von S_n und seinem eigenen Zwischenwert sc_i;
12 **end**
13 ;

Algorithmus 24 : Protokoll 2 aus [91]: Privatsphäre schützende Schlüsselvereinbarung.

Auswertung der Nutzbarkeit Algorithmus k-Means nach [96]

Im Folgenden werden die evaluierten Laufzeiten des in [96] erarbeiteten und in Kapitel 5.4.1 beschriebenen k–Means-Verfahrens auf pseudonymisierten Daten gelistet. Die Evaluation wurde im Rahmen der Abschlussarbeit [96] auf der MovieLens-Datensammlung [72] mit einem Rechner

Eingabe : Online-Dienst $\mathcal{OD}$ möchte den vom Leak betroffenen Nutzer mit dem Datum e warnen.
Ausgabe : Online-Dienst $\mathcal{OD} \in \{\mathcal{OD}_1, \cdots, \mathcal{OD}_n\}$ fragt beim Mediator $\mathcal{M}$ an, ob ein Betroffener mit der pseudonymisierten E-Mail-Adresse $c := c(e)$ bereits gewarnt wurde. $\mathcal{M}$ teilt mit, ob der Betroffene gewarnt wurde ohne dessen Identität zu erfahren.

1 $\mathcal{OD}$ berechnet und speichert $(c := c(e) := SHA3_\kappa(e), e)$;
2 $\mathcal{OD}$ berechnet $c_{\mathcal{BF}}$ als Kodierung des Einfügens von $SHA3_\kappa(e)$ in einen leeren Bloom-Filter $\mathcal{BF}_{empty}$;
3 $\mathcal{OD}$ sendet die Kodierung $c_{\mathcal{BF}}$ an $\mathcal{M}$;
4 $\mathcal{M}$ überprüft den vorliegenden Bloom-Filter auf Vorliegen von $c_{\mathcal{BF}}$. Das Ergebnis ist äquivalent zur Ausführung der Überprüfung $\mathcal{BF}.check(SHA3_\kappa(e))$;
5 **if** $\mathcal{BF}.check(SHA3_\kappa(e)) == 0$ **then**
6 | $\mathcal{M}$ sendet 0 an $\mathcal{C}$;
7 | $\mathcal{OD}$ warnt e;
8 | $\mathcal{M}$ nimmt Kenntnis davon, dass $SHA3_\kappa(e)$ einen zu warnenden Nutzer betrifft und fügt $c_{\mathcal{BF}}$ in den Bloom-Filter ein. Dies entspricht dem Ergebnis der Ausführung von $\mathcal{BF}.insert(SHA3_\kappa(e))$;
9 **end**
10 ;
11 **if** $\mathcal{BF}.check(SHA3_\kappa(e)) == 1$ **then**
12 | $\mathcal{M}$ sendet 1 ano $\mathcal{OD}$;
13 | $\mathcal{OD}$ warnt den Nutzer mit dem Datum e nicht und verwirft $(c := c(e) := SHA3_\kappa(e), e)$;
14 **end**
15 ;

Algorithmus 25 : Protokoll 3 aus [91]: Privatsphäre wahrende Überprüfung, ob ein Betroffener eines Leaks bereits von einem anderen Online-Dienst gewarnt wurde.

mit dem Betriebssystem Windows 10, Python Version 3.4.4 und einer Intel i5-7500 CPU und 8GB RAM durchgeführt. Für die Gewichte wurden Zufallszahlen der Größe 2^{128} und Paillier-Schlüssel der Länge 2048 Bit generiert. Das Abbruchkriterien wurde auf $\epsilon := 1$ festgelegt. Die Struktur der Datensammlung ist in Tabelle 2 gelistet. Die Laufzeit der Durchführung eines k–Means-Clusterings für unterschiedlich große Teilmengen der Datensammlung und unterschiedliche Werte für k ist in Tabelle 3 zusammengefasst.

Tabelle 2: *Die Datenattribute der MovieLens-Datensammlung [72].*

userId	movieId	rating	timestamp
userId_1	movieId_1	rating_1	timestamp_1
userId_2	movieId_2	rating_2	timestamp_2

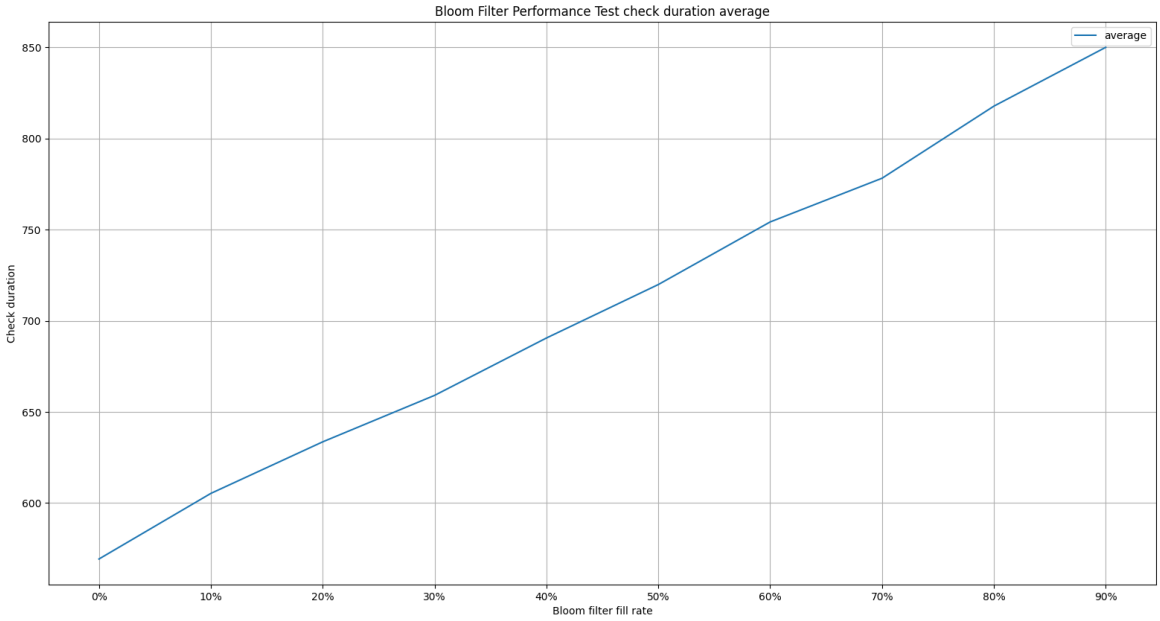

Abbildung 3: *Laufzeit einer Abfrage in Millisekunden für 50000 Klartextdaten nach Protokoll 25 in Abhängigkeit der Füllrate.*

Tabelle 3: *Laufzeiten in Sekunden für variable Anzahl der Eingabedaten, Dimensionen der Eingabedaten und Anzahl der Cluster.*

Anzahl Eingabedaten	Dimension	$k = 2$	$k = 3$	$k = 5$
10.000	10D	2.190	3.383	5466
10.000	5 D	1200	1787	2988
10.000	2 D	613	898	1506
5.000	10D	1100	1642	2730
5.000	5 D	598	926	1496
5.000	2 D	300	449	748
2.500	10D	550	822	1369
2.500	5 D	300	449	748
2.500	2 D	151	224	374

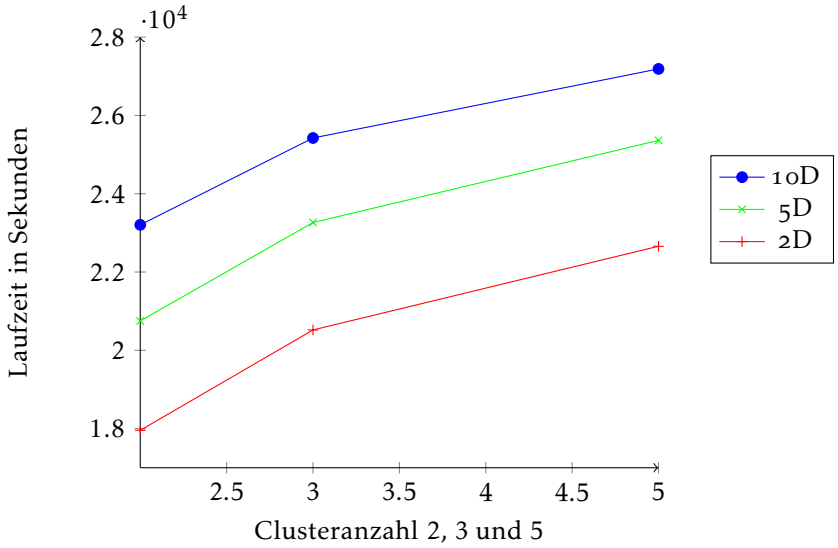

Abbildung 4: *Laufzeitenfaktor für* 2, 3 *und* 5 *Cluster und zwei-, fünf- und zehndimensionalen Eingabedaten.*

Vergleich des k–Means-Clustering-Verfahrens auf pseudonymisierten und Klartextdaten.

In Abbildung 4 zeigt einen Vergleich der Laufzeiten des Verfahrens mit einer Implementierung des herkömmlichen k–Means-Verfahrens derselben Konfiguration auf den der Pseudonymisierung zugrundeliegenden Klartextdaten zusammengefasst. Die Laufzeit des k–Means-Verfahrens auf Klartextdaten ist wesentlich geringer als das in [96] vorgestellt Verfahren. Für eine Clusterzahl von $k = 5$ und hochdimensionale Eingabedaten steigt die Laufzeit bis um den Faktor 270000 an.

Benutzeroberfläche für die Formulierung von Nutzbarkeitspolitiken in `Util` nach [148]

Um Anwender bei der Formulierung einer maschinenlesbaren Nutzbarkeitspolitik in `Util` zu unterstützen, kann eine grafische Nutzeroberfläche (Graphical-User-Interface, GUI) genutzt werden. Ein Beispiel hierfür wurde innerhalb einer Abschlussarbeit [148] erarbeitet, die im Rahmen der Forschungsarbeiten zu dieser Dissertation betreut wurde. Abbildung 5 zeigt eine solche GUI. Zu beachten ist, dass die GUI über den k-Means-Algorithmus verschiedene Anforderungen der Anforderungsklasse Algorithmus implementiert.

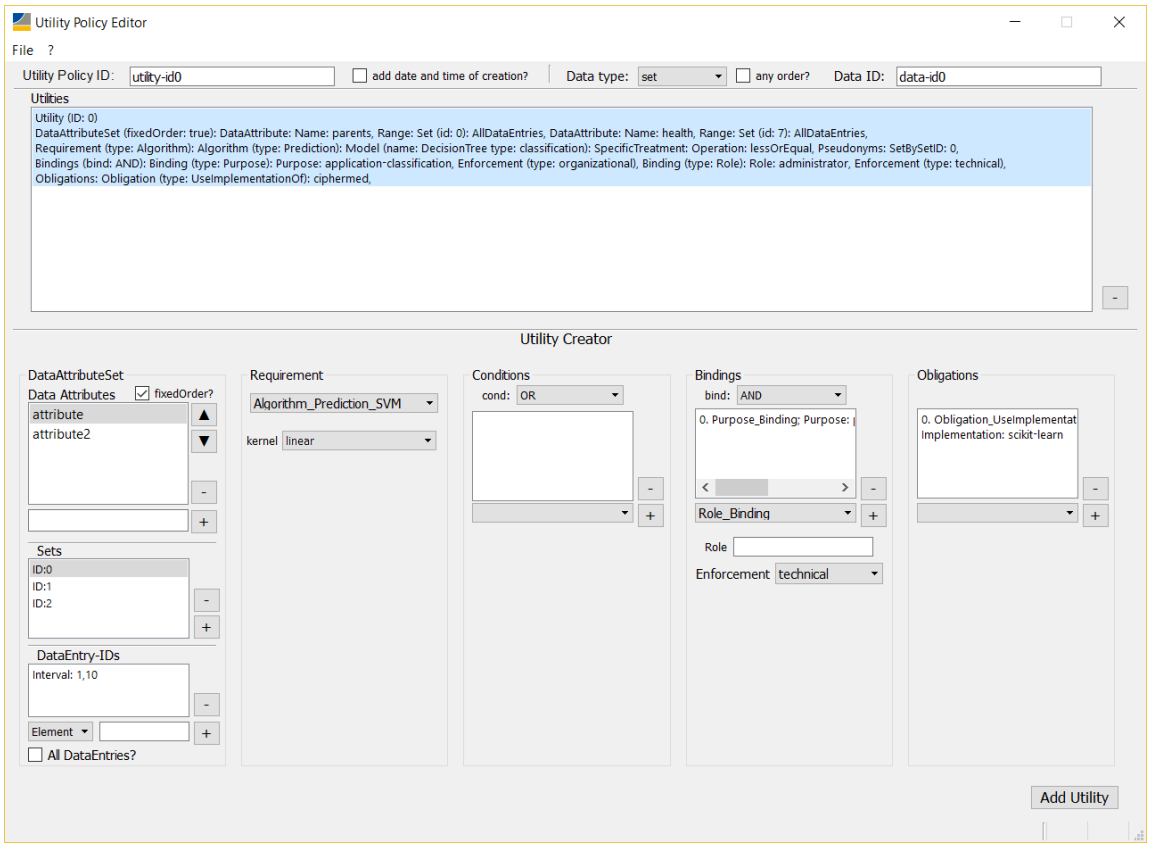

Abbildung 5: *GUI für die Unterstützung der Formulierung einer Nutzbarkeitspolitik in `Util` nach [148].*