De-Anonymisierungsverfahren: Kategorisierung und Anwendung für Datenbankanfragen

De-anonymization: Categorization and use-cases for database queries

Johannes Goltz, Hannes Grunert, and Andreas Heuer

Universität Rostock, Lehrstuhl für Datenbank- und Informationssysteme, Institut für Informatik, 18051 Rostock

Abstract: The project PArADISE deals with activity and intention recognition in smart environments. This can be used in apartments, for example, to recognize falls of elderly people. While doing this, the privacy concerns of the user should be kept. To reach this goal, the processing of the data is done as close as possible at those sensors collecting the data. Only in cases where the processing is not possible on local nodes the data will be transferred to the cloud. But before transferring, it is checked against the privacy concerns using some measures for the anonymity of the data. If the data is not valid against these checks, some additional anonymizations will be done.

This anonymization of data must be done quite carefully. Mistakes might cause the problem that data can be reassigned to persons and anonymized data might be reproduced. This paper gives an overview about recent methods for anonymizing data while showing their weaknesses. How these weaknesses can be used to invert the anonymization (called de-anonymization) is shown as well. Our attacks representing the de-anonymization should help to find weaknesses in methods used to anonymize data and how these can be eliminated.

Zusammenfassung: Im Projekt PArADISE sollen Aktivitäts- und Intentionserkennungen in smarten Systemen, etwa Assistenzsystemen in Wohnungen, so durchgeführt werden, dass Privatheitsanforderungen des Nutzers gewahrt bleiben. Dazu werden einerseits Auswertungen der Daten sehr nah an den Sensoren, die die Daten erzeugen, vorgenommen. Eine Übertragung von Daten in die Cloud findet nur im Notfall statt. Zusätzlich werden aber vor der Übertragung der nicht vorausgewerteten Daten in die Cloud diese auf Privatheitsanforderungen hin geprüft, indem Anonymisierungsmaße getestet und eventuell weitere Anonymisierungen von Daten vorgenommen werden.

Diese Anonymisierung von Datenbeständen muss mit großer Sorgfalt geschehen. Fehler können sehr schnell dazu führen, dass anonymisierte Datenbestände wieder personalisiert werden können und Daten, die eigentlich entfernt wurden, wieder zurückgewonnen werden können. Dieser Artikel betrachtet aktuelle Verfahren zur Anonymisierung und zeigt Schwachstellen auf, die zu Problemen oder gar der Umkehrung der Methoden führen können. Unsere künstlich erzeugten Angriffe durch De-Anonymisierungen sollen helfen, Schwachstellen in den Anonymisierungsverfahren zu entdecken und zu beheben.

Keywords: Datenbanken, Datenschutz, (De-)Anonymisierung

In: M. Leyer (Ed.): Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.
Rostock, Germany, 11.-13. September 2017, published at http://ceur-ws.org

1 Einleitung

Datenschutz wird in der heutigen Gesellschaft zunehmend wichtiger. Durch neuartige Techniken werden immer mehr Systeme ins Leben eingebunden, die Informationen von ihren Nutzern sammeln und auswerten. Die Auswertung kommt neben dem Nutzer auch den Anbietern dieser Softwaresysteme zugute, da sie immer einfacher und detaillierter Informationen über ihre Nutzer sammeln können, um einen besseren Service anzubieten. Für Nutzer dieser Systeme wird es hingegen zunehmend schwieriger zu erkennen, welche Daten konkret gesammelt und wie diese weiter verarbeitet werden. Zudem wird häufig mit einer Anonymisierung der Daten geworben, wobei allerdings detaillierte Informationen zur Umsetzung häufig nicht zu finden sind.

Erschwerend kommt hinzu, dass Nutzer häufig die Datenschutz- oder Nutzungsvereinbarung aus Gründen der Bequemlichkeit nicht mehr lesen und so keine Ahnung haben, wie ihre Daten weiterverarbeitet werden. Beispielsweise versprachen einige Nutzer in einem Experiment durch die Nutzung eines öffentlichen Hotspots ihr erstgeborenes Kind oder liebstes Haustier dem Hoster des Hotspots [12]. Dies zeigt, dass grundsätzlich eine große Diskrepanz zwischen der Aufklärungspflicht des Anbieters und der Bereitschaft der Nutzer, diese zu lesen, besteht.

Bei Nutzung von aktuellen Anonymisierungsverfahren muss allerdings die Implementierung genau betrachtet werden, da kleine Fehler fatale Auswirkungen auf das Resultat haben können. Zu eng gewählte Randbedingungen für eine Anonymisierung von einem Datenbestand können beispielsweise dazu führen, dass die ursprünglichen Daten rekonstruiert werden können, oder zumindest teilweise wieder personenbeziehbare Daten offengelegt werden.

Konkret soll die De-Anonymisierung vor allem für das PArADISE-Framework[1] [7] betrachtet werden. Dieses wird im folgenden Kapitel des Beitrags vorgestellt. Im darauffolgenden Kapitel 3 werden zuerst verschiedene Verfahren und Maße vorgestellt, um eine Anonymisierung quantifizierbar zu machen. Anschließend werden im Kapitel 4 unterschiedliche Varianten der De-Anonymisierung aufgezeigt, die zugleich die Probleme der einzelnen Verfahren verdeutlichen. Nach einem Kapitel zu einer möglichen Automatisierung eines Angriffes liefert die Zusammenfassung einen Überblick und Ausblick. Eine Langfassung der Thematik ist in [5] zu finden.

2 Das PArADISE-Projekt

Die Forschung an der Universität Rostock beschäftigt sich unter anderem interdisziplinär mit Assistenzsystemen. Hierbei sollen zum Beispiel Stürze in Wohnungen erkannt werden. Es wird neben der Sensorik auch die Datenverarbeitung untersucht. An dieser Stelle kommt PArADISE zum Einsatz,

[1] Privacy Aware Assistive Distributed Information System Environment (PArADISE)

welches die Prinzipien von *Privacy by Design* umsetzt, indem die Implementierung von datenschutzfördernden Techniken (Privacy Enhancing Technologies, PETs) erfolgt. Dabei werden im Speziellen die rechtlichen Anforderungen nach Datensparsamkeit und Datenvermeidung durch Techniken zur *Anfrageumschreibung* umgesetzt. Ausgehend von dem reduzierten Datenbestand werden verschiedene *Anonymisierungstechniken* verwendet, um das Ergebnis der Anfrage datenschutzkonform zu veröffentlichen. In diesem Artikel wird beschrieben, wie durch De-Anonymisierungstechniken überprüft wird, ob der scheinbar anonymisierte Datensatz wieder deanonymisiert werden kann. Ziel der Überprüfung ist die Reduzierung von Angriffsmöglichkeiten innerhalb der Verarbeitungskette im PArADISE-Framework.

Privacy by Design durch Anfrageumschreibung

Die Auswertung der Rohdaten erfolgt über SQL-Anfragen, wobei ein Schichtsystem aus logischen Schichten implementiert wurde. In Abbildung 1 ist dies gezeigt. Die verfügbaren Geräte zur Auswertung werden nach Leistungsfähigkeit in unterschiedliche Schichten eingeteilt und Daten zwischen den Schichten werden nur weiter gereicht, wenn die aktuelle Schicht nicht ausreichend Leistung zur Durchführung der Anfrage besitzt. Die Anfrage wird dabei aufgespalten und in mehrere Teilanfragen zerlegt. Jeder Knoten führt die für ihn maximal mögliche Teilanfrage aus und reicht den für ihn nicht ausführbaren Teil weiter. Auf diese Art und Weise ist es beispielsweise möglich, dass bereits Sensoren einfache Selektionen oder auch Aggregate über die letzten Werte berechnen und lediglich das Ergebnis weiterreichen. Es ist zu beachten, dass im Ergebnis deutlich weniger Informationen enthalten sind als im Originaldatenbestand aller ausgewerteten Sensoren. Daher kann Datensparsamkeit auf diese Art sehr gut umgesetzt werden. Die Anfrageumschreibung des PArADISE-Projekts ist in [8] detaillierter beschrieben.

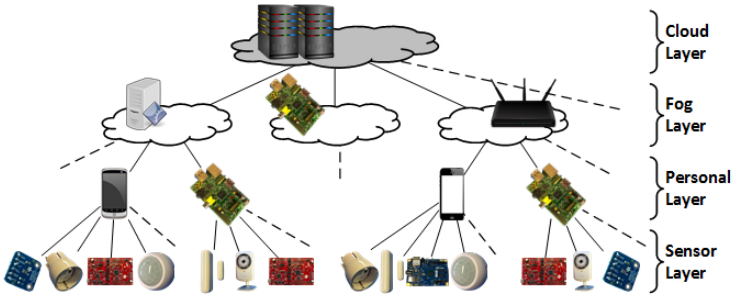

Fig. 1. Schichtaufbau des PArADISE-Frameworks

Sollten Daten an höhere Schichten weitergegeben werden, so werden diese zusätzlich mit den hinterlegten Richtlinien verglichen. Sobald zu viele Informationen enthalten sind, wird eine Anonymisierung durchgeführt.

Dazu müssen wir einerseits die zu kontrollierenden Anonymitätsmaße und ihre Parameter festlegen, andererseits aber auch Verfahren implementieren, die dieses Anonymitätsmaße auf dem in die Cloud zu übertragenden Datenbestand effizient berechnen können (siehe folgendes Kapitel 3). Um zu testen, wie sicher die Anonymität des Nutzers gewährleistet ist, entwickeln wir gleichzeitig Angriffsverfahren (De-Anonymisierung), die Schwachstellen in der Anonymisierung aufdecken sollen (siehe Kapitel 4).

3 Anonymisierungsverfahren und -maße

Um die Anonymisierung von Datenbeständen automatisieren zu können, werden entsprechende Maße benötigt, die den aktuellen Grad der Anonymität bestimmen. Sollten die vorliegenden Daten noch nicht anonym genug sein, können Algorithmen genutzt werden, um den Informationsgehalt zu verringern. Dies wird so lange iterativ in Schritten durchgeführt, bis ein entsprechendes Maß erfüllt ist. Dieser Absatz beschreibt entsprechende Methoden zum Messen des Grades der Anonymisierung. Ein Kern-Bestandteil ist dabei der Quasi-Identifikator [1].

Definition 1 *Ein Quasi-Identifikator (QI) Q_T ist eine endliche Menge von Attributen $\{A_i, \ldots, A_j\}$ einer Tabelle T mit einer endlichen Menge von Attributen $\{A_1, A_2, \ldots, A_n\}$. Hierbei gilt $\{A_i, \ldots, A_j\} \subseteq \{A_1, A_2, \ldots, A_n\}$. Mit Hilfe des QIs ist es möglich, mindestens ein Tupel der Tabelle T eindeutig zu bestimmen [9]. Eine Menge von Tupeln t von T, welche bezüglich des QIs Q_T nicht unterscheidbar sind, wird als q^*-Block bezeichnet.*

Innerhalb von PArADISE werden QIs für die Parametrisierung der verschiedenen Komponenten und Algorithmen, wie dem Modul zur Generierung von Datenschutzeinstellungen und dem Präprozessor zur Reformulierung von Anfragen, genutzt (siehe Abbildung 2). Speziell im Postprozessor werden anhand von QIs die Anonymitätsmaße überprüft. Durch die vorherige Projektion der Attributmenge müssen nicht alle Attribute zum Finden von QIs in Betracht gezogen werden. Durch den in [6] vorgestellten Algorithmus können die minimalen QIs effizient berechnet werden. Ausgehend von der Höhe des erwarteten Informationsverlustes für eine gegebene Anfrage wird dasjenige Anonymisierungsverfahren, welches gleichzeitig eine hohe Anonymisierung als auch einen geringen Informationsverlust bietet, ausgewählt.

Ein sogenanntes „sensitives Attribut“ ist ein Attribut, das nicht mit personenbeziehbaren Informationen in Verbindung gebracht werden darf, da dies der entsprechenden Person schaden könnte. Beispielsweise könnte dieses Attribut die Diagnose in einer Tabelle sein, in der Patientendaten mit entsprechenden Diagnosen abgespeichert sind (siehe Tabelle 1). Während die Informationen der

Spalte *Diagnose* allein nicht problematisch sind, werden sie in Verbindung mit Name und Vorname durchaus kritisch. Die Mengen der sensitiven Attribute und der Attribute von QIs und Schlüsseln sind nicht zwangsweise disjunkt. Es kann daher vorkommen, dass jedes Attribut Teil eines QIs ist.

3.1 Anonymisierungsmaße

Die im Folgenden vorgestellten Maße für die Anonymität einer Relation lassen sich vor allem in Kombination mit der Technik der *Generalisierung* und *Unterdrückung* einsetzen. Diese werden im weiteren Verlauf vorgestellt.

k-Anonymität

Die *k-Anonymität* stellt die geringsten Anforderungen an die zu bewertenden Daten. Der Wert k gibt dabei an, wie viele Tupel es mit jeweils gleichem QI geben muss. Eine formale Definition ist in [9] zu finden.

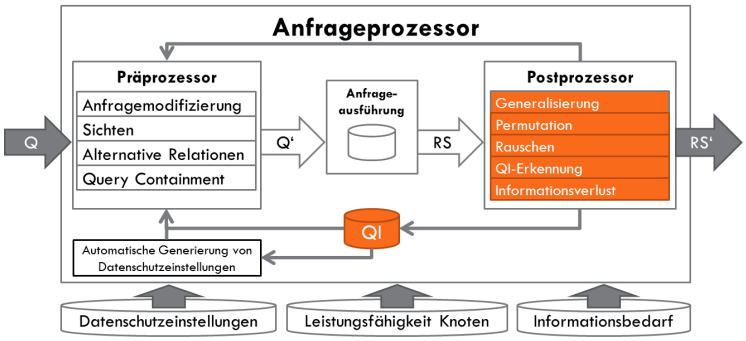

Fig. 2. Einordnung der Deanonymisierung in das PArADISE-Framework

Je nach Wert für k und dem QI müssen die Daten, sollten sie aktuell nicht die geforderte k-Anonymität erfüllen, verallgemeinert werden. Dafür kann sehr gut die Generalisierung genutzt werden. Der Vorgang wird dabei iterativ so lange wiederholt, bis eine ausreichende Anonymisierung durchgeführt wurde. Beispielhaft ist dies in Tabelle 1 gezeigt.

l-Diversität und t-Closeness

l-Diversität und *t-Closeness* stellen Verschärfungen der k-Anonymität dar. *l-Diversität* nimmt sich der Problematik an, dass der Attributwert des sensitiven Attributes eines q^*-Blocks für jedes Tupel darin gleich sein könnte. Angenommen der Attributwert *Röteln* sei in Tabelle 1 ebenfalls *Diabetes*, dann würde die Tabelle damit immer noch k-Anonymität für k=2 erfüllen, allerdings keine

l-Diversität für *l=2* mehr. Der Wert l gibt entsprechend an, wie viele unterschiedliche Werte für das sensitive Attribut im entsprechenden q^*-Block auftauchen müssen [9].

Bei *t-Closeness* wird die Verteilung der Attributwerte des sensitiven Attributes in Bezug zur Verteilung der Attributwerte in der gesamten Relation betrachtet. Die Verteilung darf dabei pro q^*-Block höchstens um t von der Gesamtverteilung abweichen [9]. Eine Herausforderung dieses Verfahrens ist die Messung der Verteilung der Werte. Während dies bei numerischen Attributwerten einleuchtend und vergleichsweise einfach erscheint, wird es bei abstrakten Werten komplizierter. Hier bieten sich die Kullback-Leibler- oder auch die Jensen-Shannon-Divergenz an [14]. Für t gilt, im Gegensatz zu k und l, je kleiner desto anonymer werden die Daten. Typischerweise liegt der Wert für t zwischen 0 und 1.

Differential Privacy

Ein weiteres Maß zum Messen der Anonymität stellt Differential Privacy [3] dar. Dabei geht es darum, ein Tupel in einer Menge von Tupeln zu schützen. Vor allem Auswertungsergebnisse sollen nicht ersichtlich machen, ob ein gewisses Tupel enthalten ist oder nicht. Unter Differential Privacy werden verschiedene Verfahren zum Hinzufügen von *Rauschen* auf den Daten und in den Anfragen zusammengefasst [4]. Vorteile ergeben sich bei Differential Privacy bei Aggregationen auf dem gesamten Datensatz, da das hinzugefügte Rauschen die Verteilung der Daten nur minimal beeinflusst. Das Rauschen führt jedoch zu Nachteilen bei der Auswertung von wenigen, aber vollständigen (d. h. alle Attribute enthaltenden) Tupeln, da jeder einzelne Attributwert verrauscht wird. Dadurch entsteht ein höherer Informationsverlust als bei der Generalisierung. Allerdings muss darauf geachtet werden, dass kein symmetrisches Rauschen eingesetzt wird, da dies von Angreifern herausgerechnet werden könnte.

3.2 Anonymisierungsverfahren

Ein typisches Verfahren zur Anonymisierung von Datenbeständen wird als *Generalisierung* bezeichnet. Es kann auch mit der *Unterdrückung* kombiniert werden. Konkrete Attributwerte werden dabei auf ein Intervall abgebildet, sodass ein Teil der Informationen verloren geht und der Grad der Anonymisierung steigt.

Generalisierung

Die *Generalisierung* ist hierbei ein spaltenorientiertes Verfahren. Es werden zu generalisierende Attribute ausgewählt, anschließend alle Attributwerte dieser Spalten (die Domäne) auf ein entsprechendes Intervall abgebildet. Die originalen Werte einer Tabelle bilden die Grunddomäne, welche auf weitere Domänen generalisiert wird [13].

Unterdrückung

Das Verfahren der *Unterdrückung* arbeitet im Gegenzug dazu auf Zeilenebene. Es kann dazu genutzt werden, um *Ausreißer* zu streichen, somit ein Anonymisierungsmaß zu erfüllen, und dabei weniger Generalisierungsschritte durchzuführen. Das Tupel, welches den Wert enthält, der unterdrückt werden soll, wird dabei komplett generalisiert. Das bedeutet, dass für alle Attributwerte ein " *" eingetragen wird. Hierbei ist es sinnvoll, eine Obergrenze an möglichen Unterdrückungen anzugeben. Ansonsten könnte es passieren, dass durch den starken Einsatz der Unterdrückung zwar eine Anonymisierung mit vergleichsweise wenig Generalisierungsschritten erreicht werden kann, allerdings sind die Daten nicht mehr repräsentativ, da zu viele Werte gestrichen wurden [13].

Zeile	Alter	Diagnose
1	13	Diabetes
2	84	Fraktur des Beins
3	20	Blutkrebs
4	28	Inkontinenz
5	12	Röteln

Zeile	Alter	Diagnose
1	10-19	Diabetes
2	*	*
3	20-29	Blutkrebs
4	20-29	Inkontinenz
5	10-19	Röteln

Table 1. Beispieltabelle (original links, generalisiert und unterdrückt rechts) - Zur Vereinfachung wurden nur zwei Spalten genutzt. Der QI sei das *Alter*, die *Diagnose* das sensitive Attribut. Durch Unterdrückung konnte der Wert der Spalte *Alter* auf ein Intervall von 10 abgebildet werden und die Tabelle erfüllt k-Anonymität für k=2 und l-Diversität für l=2 (q^*-Blöcke wurden farblich hervorgehoben).

In der rechten Relation von Tabelle 1 ist zu erkennen, wie Generalisierung und Unterdrückung arbeiten. Zeile 2 wurde unterdrückt, da das Alter einen stark abweichenden Wert im Verhältnis zu den anderen Werten darstellt. Um trotz des Wertes k-Anonymität für k=2 zu erfüllen, hätten die Werte sonst auf ein entsprechend großes Intervall abgebildet werden müssen und alle Werte hätten mit großer Wahrscheinlichkeit im selben Intervall gelegen. Die einzelnen q^*-Blöcke wurden zusätzlich farblich hervorgehoben. Sie unterscheiden sich bezüglich des QIs nicht. Ist bekannt, welches Alter die entsprechende Person hat, so ist nicht mehr ersichtlich, welche Diagnose ihr gestellt wurde.

Slicing

Ein weiteres Verfahren wird als Slicing bezeichnet. Hierbei wird eine Relation R in m vertikale und n horizontale Teilrelationen aufgeteilt. Innerhalb dieser Teilrelationen werden die Tupel zufällig sortiert, bevor alle Teilrelationen wieder zu einer kompletten Relation zusammengefügt werden [10]. Es ist zu beachten, dass unbedingt angegeben werden muss, an welchen Stellen in der Relation die Trennung vorgenommen wurde. Zusammenhängende Auswertungen zwischen Attributen, die in unterschiedlichen Teilrelationen standen, sind nicht mehr

möglich, da die Reihenfolge unabhängig und zufällig verändert wurde. Zwischen Attributen, die gemeinsam in einer Teilrelation standen, kann allerdings problemlos eine Auswertung stattfinden, da die Zusammenhänge nicht verändert wurden. Mit der Technik ist es somit möglich, trotz äußerst geringem Informationsverlust eine gute Anonymisierung zu erreichen. Natürlich muss sehr genau beachtet werden, zwischen welchen Attributen die Tabelle aufgetrennt wird.

4 De-Anonymisierungsverfahren

Wir beschreiben nun, an welchen Stellen die vorher vorgestellten Anonymisierungsverfahren versagen. Es lassen sich grundsätzlich zwei unterschiedliche Ansätze unterscheiden. Zum einen kann lediglich die Anfrage zur De-Anonymisierung von Daten betrachtet werden, zum anderen ist auch eine De-Anonymisierung auf Grundlage der vorliegenden Daten möglich.

4.1 Anfragebasierte De-Anonymisierungsverfahren

Datenbankmanagementsysteme bieten die Möglichkeit, den Zugriff komplett auf einzelne Sichten zu beschränken. Diese Technik kann eingesetzt werden, um den Zugriff nur auf ganz bestimmte Attribute zu erlauben. Das Besondere ist, dass auch Attributkombinationen veröffentlicht werden können, die ohne Joins nicht abzufragen sind, wobei in der Ergebnisrelation der Sicht das verbindende Attribut ausgeblendet wird und nicht eingesehen werden kann. Anfragen, welche an das DBMS gestellt werden, die auf die internen Relationen zugreifen, können automatisiert in Anfragen umgeschrieben werden, die lediglich Sichten einsetzen. Hierzu wird die sogenannte Answering-Queries-using-Views-Technik [2] eingesetzt. Damit ist es möglich, Anfragen automatisiert umzuschreiben. Sollte keine gleichwertige Anfrage mit den Sichten erreicht werden können, so wird eine Anfrage formuliert, die ein Maximum der Antwort enthält, die mit den originalen Tabellen möglich wäre. Allerdings sind die Algorithmen derzeit noch nicht in der Lage, sehr komplexe SQL-Operationen umzuformulieren. Diese wären allerdings nötig, um Machine-Learning-Algorithmen umzusetzen, die in der Entwicklung von Assistenzsystemen beispielsweise zur Aktivitäts- und Intentionserkennung eingesetzt werden.

Da wir uns in diesem Artikel schwerpunktmäßig mit der datenbasierten De-Anonymisierung befassen, verweisen wir für Details zu den bekannten Verfahren und unsere Weiterentwicklung in Richtung einer Answering-Queries-using-Operators-Technik auf [8]. Sollte diese Technik jedoch eingesetzt werden, und alle Anfragen entsprechend auf erlaubten Sichten arbeiten oder entsprechend umformuliert werden können, muss auf jeden Fall die Gesamtheit der Sichten auf Schwachstellen betrachtet werden. Insbesondere muss geprüft werden, ob es nicht möglich ist, zwischen verschiedenen Ergebnisrelationen durch entsprechende Selektionsbedingungen Joins durchführen zu können, die dazu führen, dass der Angreifer Informationen verknüpfen kann, welche nicht in direkten Zusammenhang gebracht werden dürfen. Zudem müssen die Ergebnisrelationen auch genau geprüft werden und sollten im Zweifel noch weiter anonymisiert werden.

4.2 Datenbasierte De-Anonymisierungsverfahren

Bei datenbasierten Verfahren wird lediglich auf die aus der Auswertung erhaltende Ergebnisrelation einer Anfrage geachtet, und nicht auf die Anfrage an sich. Hier kommen die im vorangegangenen Abschnitt vorgestellten Anonymisierungsmaße zum Einsatz, um den Grad der Anonymität zu bestimmen. Diese weisen allerdings Schwachstellen auf, die Beachtung finden müssen.

K-Anonymität bildet das Anonymisierungsmaß mit der geringsten Anforderung, daher sind auch hier besonders einfach Schwachpunkte zu finden. Ein großes Problem stellt die Selektivität des sensitiven Attributes dar [11]. Sollte ein q^*-Block k-Anonymität entsprechend der Anforderungen erfüllen, kann es jedoch passieren, dass das sensitive Attribut aller Tupel in diesem Block den selben Wert annimmt. Dies ist problematisch, da so die Daten ohne aktives Zutun des Angreifers aufgrund der Homogenität extrahiert werden können. Beispielhaft ist dies in Tabelle 1 zu sehen.

Das Maß der l-Diversität nimmt sich dieses Problems teilweise an. Der Wert von l gibt an, wie viele unterschiedliche Attributwerte innerhalb eines q^*-Blocks vorkommen müssen. Je nach Unterschied der Werte kann dies allerdings noch immer problematisch sein. Als Beispiel sollen Krankheiten dienen. Es kann sein, dass für das sensitive Attribut eines Blocks lediglich unterschiedliche Krebsarten vorkommen, dies allerdings für den Angreifer bereits ausreichend viele Informationen sind. Tabelle 1 zeigt dieses Problem. Sobald bekannt ist, dass die Person zwischen 20 und 29 Jahren alt ist und in der Tabelle vorkommt, kann abgeleitet werden, dass sie eine Art von Krebs hat. Eine deutlich bessere Lösung des Problems bietet das Maß der t-Closeness. Hierbei wird auch die Verteilung der Werte im sensitiven Attribut innerhalb eines q^*-Blocks in Bezug zur Verteilung der Werte innerhalb der gesamten Relation betrachtet. Dabei darf ein Schwellwert t nicht überschritten werden. Bei restriktiver Anwendung kann diese Problematik mit sehr hoher Wahrscheinlichkeit eliminiert werden.

Ein ähnlich gelagertes Problem stellt gutes Hintergrundwissen dar. Problematisch wird dies vor allem bei einer Anonymisierung von Daten, die keinen strengen Anforderungen an k-Anonymität und l-Diversität stellt [11]. Es sind immer genau *x-1* Fakten notwendig, um ein Tupel aus einer Gruppe von x Tupeln eindeutig zu identifizieren. Durch den Einsatz von t-Closeness kann das Problem gemildert werden, da die Verteilung der Werte für das sensitive Attribut ähnlich zur gesamten Relation ist. Allerdings ist auch damit eine Identifizierung durch Hintergrundwissen nicht ausgeschlossen.

Je nach veröffentlichten Daten kann auch die Sortierung der Tupel dem *Angreifer* helfen, persönliche Daten aus Ergebnissen zu extrahieren. Grundsätzlich sind Ergebnisrelationen immer sortiert. Dies liegt an den internen Speicherstrukturen der Datenbanksysteme [15]. Sollten allerdings mehrere Veröffentlichungen der gleichen Daten mit unterschiedlichen Quasi-Identifikatoren gemacht werden, so kann es zum Problem kommen, dass diese Daten eventuell einfach über die Sortierung verknüpft werden können. In Tabelle 2 ist dies beispielhaft zu sehen. Ähnlich verhält es sich, wenn der Angreifer einen direkten Zugang zur Datenbank nutzen kann. Damit könnte er die gleiche Anfrage mehrfach stellen und so hoffen,

dass vom System unterschiedliche Attribute der Quasi-Identifikatoren gewählt werden und so die Anonymisierung unterschiedlich umgesetzt wird. Zusätzlich könnte es auch passieren, dass eventuell ein anderer Quasi-Identifikator gewählt wird. Das Problem lässt sich allerdings auch sehr leicht beheben, indem die Ergebnisrelation einfach vor der Veröffentlichung zufällig sortiert wird.

Geburtsjahr	Postleitzahl
1994	18055
1983	18057
1965	18055
1963	18055
1975	18059
1977	18057
1955	18181

Geburtsjahr	Postleitzahl
1980-1994	18055
1980-1994	18057
1965-1979	18055
1950-1964	18055
1965-1979	18059
1965-1979	18057
1950-1964	18181

Geburtsjahr	Postleitzahl
1994	18000-18199
1983	18000-18199
1965	18000-18199
1963	18000-18199
1975	18000-18199
1977	18000-18199
1955	18000-18199

Table 2. Ursprungstabelle (links) und jeweils eine der Spalten anonymisiert, sodass k-Anonymität für k=2 erfüllt ist. Quasi-Identifikator ist *Geburtsjahr* und *Postleitzahl*. Die Originaltabelle lässt sich durch direktes nebeineinanderlegen rekonstruieren.

Bei mehreren Veröffentlichungen der gleichen Daten muss darauf geachtet werden, dass immer der gleiche Quasi-Identifikator gewählt wird, oder zumindest alle Attribute, die im Quasi-Identifikator der ersten Veröffentlichung enthalten waren, im Neuen auch enthalten sind. Ansonsten ist es einem *Angreifer* unter Umständen möglich, durch die wechselnden Attribute Joins über den Veröffentlichungen zu erstellen und somit private Daten zu rekonstruieren [15]. Ähnlich verhält es sich bei zeitlich versetzten Veröffentlichungen. Hier muss geprüft werden, wie sich die beiden Veröffentlichungen unterscheiden. Sollte durch die Änderung des Datenbestandes eine geringere Generalisierung stattfinden, könnte es dazu kommen, dass Informationen genauer spezifiziert werden können, als es mit der ursprünglichen Veröffentlichung möglich war.

5 Automatisierung eines Angriffs

Besonders wünschenswert ist für einen Angreifer natürlich eine vollständige Automatisierung des Angriffs. Dies hilft aber nicht nur dem späteren Angreifer, sondern in der Entwicklungsphase bereits dem Entwickler des Assistenzsystems, der das Prinzip *Privacy by Design* realisieren und Schwachstellen aufdecken möchte. Für anfragebasierte und datenbasierte De-Anonymisierungen wollen wir daher auch Methoden entwickeln, um Angriffe automatisch zu generieren — und diese danach durch Verschärfung der Anonymisierungsmaße und Verschärfung der erlaubten Sichten zu verhindern.

Dies würde sehr viel Arbeit ersparen, ist aktuell aber nur mit äußerst großem Aufwand realisierbar. Eine Hilfestellung für die Wahl des richtigen Angriffsvektors hingegen kann durch vergleichsweise einfache Techniken erreicht werden.

Durch eine statistische Auswertung der Ergebnisse kann ein schneller Überblick über die vorliegenden Daten gewonnen werden.

Hilfreich ist zudem das Suchen nach vorhandenen Quasi-Identifikatoren im Ergebnis der Anfrage, da diese eine Kombination von Attributen darstellen, die besonders selektiv sind. Hierzu bietet sich vor allem der *TopDownBottomUp*-Ansatz an (siehe [6]). Dabei werden alle, und vor allem auch minimale, Quasi-Identifikatoren gefunden. Ein minimaler Quasi-Identifikator zeichnet sich dadurch aus, dass es keinen weiteren Quasi-Identifikator gibt, der aus weniger Attributen besteht. Dies führt dazu, dass ein Angreifer lediglich ein Minimum an Informationen sammeln muss, um beispielsweise mittels Hintergrundwissen wieder auf persönliche Daten zurück schließen zu können. Unser Angriff wurde unauffälliger, indem wir die Auswertung lediglich auf der lokalen Kopie des Anfrageergebnisses ausgeführt haben und wir somit keine zusätzlichen Abfragen an die Datenbank stellen mussten. Auf diese Art und Weise ist es einem Angreifer möglich, einen schnellen Überblick über die abgefragten Daten zu gewinnen und damit das weitere Vorgehen entsprechend zu steuern oder den Aufwand einer Deanonymisierung einzuschätzen.

Sollten Werte, welche für die Bestimmung der statistischen Daten benötigt werden, aus der Datenbank abgefragt werden, könnte es zu Problemen kommen, wenn sich in der Zwischenzeit der Datenbestand verändert hat, oder auch die Ausgabe für jede Anfrage eventuell anders anonymisiert wurde. Weiterhin wurde in PArADISE eine Möglichkeit geschaffen, Wertebereiche der einzelnen Spalten einschränken zu können, um so fehlerhafte beziehungsweise nicht relevante Werte aus der statistischen Berechnung ausschließen zu können (siehe [5]).

Durch eine automatisierte Generalisierung können die Attributwerte des sensitiven Attributes so lange iterativ generalisiert werden, bis für jeden q^*-Block ein eindeutiger Wert zugeordnet ist. Dabei müssen Duplikate nach jeder Iteration gelöscht werden. Mit den Informationen ist es einem Angreifer anschließend möglich, einen allgemeineren, aber immer noch möglichst spezifischen, Wert zu erkennen, ohne dass er aktiv einschreiten muss.

6 Zusammenfassung

Grundsätzlich lässt sich sagen, dass trotz Anonymisierung die Daten nie zu 100 Prozent sicher vor einem Angriff sind. Allerdings kann die Möglichkeit der De-Anonymisierung durch Angreifer sehr stark verringert werden. Auf der anderen Seite muss geprüft werden, ob die anonymisierten Daten noch für die nötigen Auswertungen ausreichend Informationen enthalten. Es sollte ein Maximum für die Werte der Anonymisierungsmaße gewählt werden, sodass gerade noch genügend Informationen für die gutartigen Anfragen enthalten sind, die von einem Assistenzsystem für die Analyse erlaubter Aktivitäten (wie Stürze) benötigt werden. Die Ausführung von bösartigen Anfragen, etwa die Ableitung genauerer Nutzerprofile oder Bewegungsprofile, die nicht zur Analyse der erlaubten Aktivitätserkennung beitragen, sollten dagegen verhindert werden.

Die fertige Lösung sollte auch schon während des Entstehungsprozesses und vor allem am Ende intensiv aus Sicht eines möglichen Angreifers betrachtet werden, um eventuelle Schwachpunkte zu lokalisieren und diese abstellen zu können. Die Answering-Queries-using-Views-Technik ist ein sehr vielversprechender Ansatz, allerdings fehlt für den produktiven Einsatz noch eine automatisierte Umschreibung von komplexeren SQL-Operationen. Hieran wird gerade im Rahmen des PArADISE-Projektes gearbeitet [8].

Das Schichtkonzept des PArADISE-Frameworks [7] bietet eine sehr gute Voraussetzung für die Anonymisierung von Daten. Es kann einfach differenziert werden, wohin die Daten weiter gereicht werden und wie stark sie entsprechend anonymisiert werden müssen. Die trotz des Schichtkonzeptes in die Cloud zu übertragenden Daten, die für den Anbieter des Assistenzsystems erforderlich sind, um die Aufgaben des Assistenzsystems erfüllen zu können, müssen dann schlussendlich mit den in diesem Artikel vorgestellten Verfahren (a) auf Anonymität geprüft, (b) eventuell weiter generalisiert und gefiltert, und (c) durch die automatische Generierung von Angriffen auf Schwachstellen geprüft werden. Durch die Kombination von anfrage- und datenbasierten Verfahren für die De-Anonymisierung hoffen wir aber, in PArADISE ein höchstmögliches Niveau an Privatheit des Nutzers bewahren zu können (siehe auch [7]).

Danksagung

Wir danken den anonymen Gutachtern für ihre konstruktiven Kommentare.

Literaturverzeichnis

1. Dalenius, T.: Finding a Needle In a Haystack or Identifying Anonymous Census Records. Journal of official statistics 2(3), 329 (1986)
2. Doan, A., Halevy, A.Y., Ives, Z.G.: Principles of Data Integration. Morgan Kaufmann (2012)
3. Dwork, C.: Differential Privacy. In: Encyclopedia of Cryptography and Security (2nd Ed.), pp. 338–340. Springer (2011)
4. Dwork, C., Roth, A.: The Algorithmic Foundations of Differential Privacy. Foundations and Trends in Theoretical Computer Science 9(3-4), 211–407 (2014)
5. Goltz, J.: De-Anonymisierungsverfahren: Kategorisierung und deren Anwendung für Datenbankanfragen. Bachelorarbeit, Universität Rostock (2017)
6. Grunert, H., Heuer, A.: Big Data und der Fluch der Dimensionalität. In: Proceedings of the 26th GI-Workshop Grundlagen von Datenbanken, Bozen-Bolzano, Italy, October 21st to 24th, 2014. pp. 29–34. `http://ceur-ws.org` (2014)
7. Grunert, H., Heuer, A.: Datenschutz im PArADISE. Datenbank-Spektrum 16(2), 107–117 (2016), `http://dx.doi.org/10.1007/s13222-016-0216-7`
8. Grunert, H., Heuer, A.: Rewriting complex queries from cloud to fog under capability constraints to protect the users' privacy. Open Journal of Internet Of Things 3(1), 31–45 (2017), proceedings of the International Workshop on Very Large Internet of Things in conjunction with the VLDB 2017 Conference in Munich, Germany.
9. Hauf, D.: Allgemeine Konzepte - K-Anonymity, l-Diversity and T-Closeness. IPD Uni-Karlsruhe (2007), zuletzt aufgerufen am 14.10.2016

10. Li, T., Li, N., Zhang, J., Molloy, I.: Slicing: A New Approach for Privacy Preserving Data Publishing. IEEE Trans. Knowl. Data Eng. 24(3), 561–574 (2012), http://dx.doi.org/10.1109/TKDE.2010.236
11. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data 1 (Mar 2007), http://doi.acm.org/10.1145/1217299.1217302
12. Melissa Michael: The Dangers of Public WiFi – And Crazy Things People Do To Use It. https://safeandsavvy.f-secure.com/2014/09/29/danger-of-public-wifi/ (2014), zuletzt aufgerufen am 13.06.2017
13. Samarati, P., Sweeney, L.: Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression. Tech. rep., Technical report, SRI International (1998)
14. Sha, C., Li, Y., Zhou, A.: On t-Closeness with KL-Divergence and Semantic Privacy. In: International Conference on Database Systems for Advanced Applications. pp. 153–167. Springer (2010)
15. Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10(05), 557–570 (2002)