Entwicklung einer neuen fehlertoleranten Methode bei der Verknüpfung von personenbezogenen Datenbanken unter Gewährleistung des Datenschutzes

Development of a New Method for Privacy-Preserving Record Linkage Allowing for Errors in Identifiers

Rainer Schnell, Tobias Bachteler und Jörg Reiher

Zusammenfassung
Die Verknüpfung der Angaben mehrerer Datenbanken über dieselbe Person wird immer häufiger für Forschungszwecke genutzt. Aus Datenschutzgründen müssen die Identifikatoren in vielen Fällen vor der Zusammenführung verschlüsselt werden. Bisher verwendete Techniken sind hierbei ineffizient, da Fälle mit Fehlern in den Identifikatoren fast immer vollständig verloren gehen. Die Autoren haben ein neues Verfahren entwickelt, das trotz starker Verschlüsselung Fehler in den Identifikatoren toleriert. Testergebnisse anhand simulierter und echter Datenbestände zeigen, dass das Verfahren ähnlich gute Ergebnisse erbringt wie unverschlüsselte Identifikatoren. Das Verfahren kann für viele Probleme in der Forschungspraxis der empirischen Sozialforschung verwendet werden.

Abstract
Combining multiple databases with additional information on the same person is increasingly occurring throughout research. In many applications, identifiers have to be encrypted due to privacy concerns. Existing protocols are inefficient in actual research practice since cases with errors in identifiers are almost always in their entirety lost. Therefore, a new protocol for privacy-preserving record linkage with encrypted identifiers allowing for errors in identifiers has been developed by the authors. The results from tests on simulated and actual databases are comparable to non-encrypted identifiers. This new technique will have many practical applications in social research.

1 Problemstellung

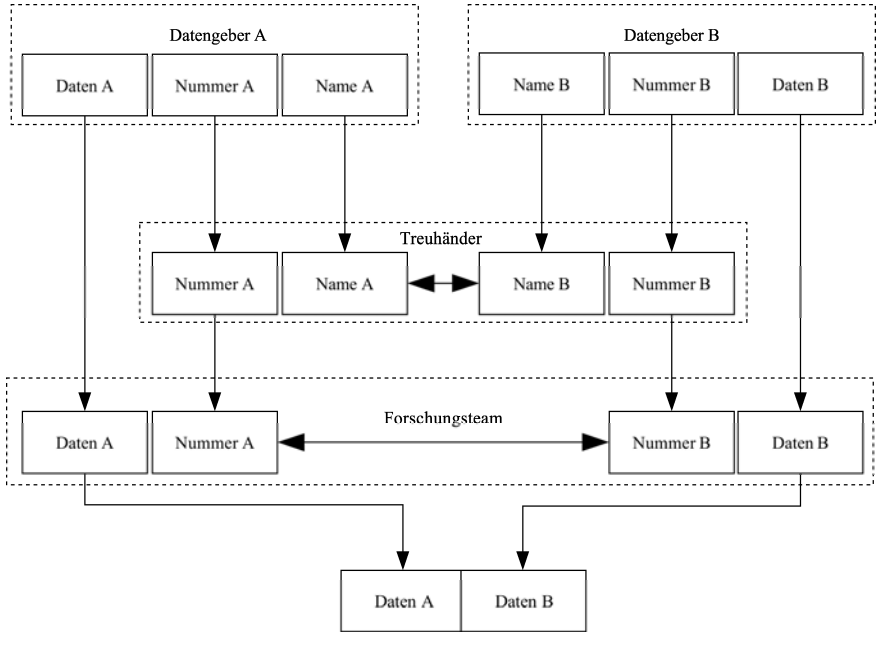
Weltweit werden in zunehmendem Maße bei Forschungsprojekten bereits existierende Datenbestände zu neuen Datenbeständen zusammengeführt (Bethlehem 2008; Schnell 2009). Dies gilt nicht nur für die hier führende Medizin, sondern auch für die Sozialwissenschaften. International üblich ist z. B. die gemeinsame Verwendung von Befragungsdaten mit Registerdaten, so bei Studien über Erkrankungshäufigkeiten, kriminologischen Fragestellungen oder Arbeitsmarktstudien. Die Zusammenführung von Datenbanken ist dann technisch unproblematisch, wenn in allen beteiligten Datenbanken eine gemeinsame Identifikationsnummer vorhanden ist, etwa eine Sozialversicherungsnummer oder – wie in den skandinavischen Ländern – eine über den Lebenslauf unveränderliche Personenkennziffer. Die meisten dieser Identifikationsnummern enthalten interne Prüfziffern, so dass Fehler in diesen Identifikatoren meist schnell entdeckt und korrigiert werden können. Probleme entstehen dann, wenn in den Datenbanken lediglich potenziell fehlerbehaftete alphanumerische Variablen wie Name, Vorname, Adresse und Geburtsort als primäre Identifikatoren vorhanden sind. Für die Zusammenführung solcher Datenbanken werden spezielle Verfahren benötigt, die zusammenfassend in der statistischen Literatur als „Record-Linkage"-Verfahren bezeichnet werden (Herzog/Scheuren/Winkler 2007). In der sozialwissenschaftlichen Forschung besteht ein zentrales Problem des Record-Linkage in der Lösung der Datenschutzprobleme: Wie können Datenbanken mithilfe von alphanumerischen Identifikatoren zusammengeführt werden, wenn die Identifikatoren Fehler aufweisen und die Anonymität der Personen in den Datenbanken vollständig gewahrt werden soll? Die Lösung dieses Problems ist Gegenstand unseres von der DFG geförderten Projekts „Spezifizierung und Implementierung eines datenschutzrechtlich unbedenklichen Verfahrens zur Verknüpfung sozialwissenschaftlicher Mikrodaten".

2 Bisherige Lösungsansätze

Um die Vertraulichkeit der Angaben in den Datenbanken zu erhalten, werden Datenbestände der beschriebenen Art meist mit Hilfe eines Datentreuhändermodells zusammengeführt. Bei einem einfachen Treuhändermodell (vgl. Abbildung 1) übermitteln die beiden Datengeber einem Datentreuhänder lediglich die zur Verknüpfung benötigten Merkmale, etwa Name, Geburtsdatum und Adresse, zusammen mit einer beliebigen, aber für den jeweiligen Datensatz ein-eindeutigen laufenden Nummer. Der Datentreuhänder verknüpft die Datenzeilen anhand der Merkmale

und erhält so Paare laufender Nummern von zusammengehörigen Records. Danach löscht der Treuhänder die Verknüpfungsmerkmale und übermittelt der Forschergruppe lediglich die Paare laufender Nummern. Die Forschungsgruppe kann so die von den Datengebern zuvor übermittelten Sachdaten zusammenführen, ohne dass die identifizierenden Merkmale bekannt sind.

Abbildung 1 Datentreuhändermodell



Quelle: Schnell, Hill & Esser (2008: 256).

In der Bundesrepublik existieren keine zentralen Datentreuhänderstellen, daher müssen die Treuhänderlösungen praktisch für jedes neue Projekt neu eingerichtet und mit Datenschützern verhandelt werden. Faktisch verhindert diese infrastrukturelle Hürde viele Projekte, da solche Genehmigungsprozesse Jahre in Anspruch nehmen können (Schnell/Bachteler 2006). Die in der Medizin in der BRD gebräuchlichen und mit beachtlichem finanziellen und personellen Aufwand betriebenen Pseudonymisierungsstrategien (Eichelberg/Aden/Thoben 2005) sind aufwendig und fehleranfällig. Aufgrund ähnlicher Probleme in anderen Ländern gibt es international eine Reihe von Forschergruppen, die Problemlösungen vorgeschlagen haben. Das bekannteste Verfahren in diesem Zusammenhang wurde von Churches und

Christen (2004) vorgestellt.[1] Aber auch dieses Verfahren besitzt zahlreiche Effizienzprobleme und Angriffsmöglichkeiten, so dass bislang kein praktisch verwendbares Verfahren existiert.[2] Es galt also, ein neues Verfahren zu entwickeln.

3 Entwicklungsprozess des neuen Verfahrens

Die Arbeitsgruppe der Autoren hat in einer Reihe von DFG-Projekten ein Programm zum Record-Linkage für die empirische Sozialforschung entwickelt: Die sogenannte „Merge Toolbox" (MTB) (Schnell/Bachteler/Bender 2004; Schnell/Bachteler/Reiher 2005). Mit MTB lassen sich Datenbanken mit probabilistischem Record-Linkage zusammenführen. MTB ist für akademische Zwecke frei verfügbar und wurde in zahlreichen empirischen Projekten erfolgreich eingesetzt. Zu diesen Anwendungen gehört der Einsatz in mehreren Krebsregistern sowie in zahlreichen Studien des Instituts für Arbeitsmarkt- und Berufsforschung der Bundesagentur für Arbeit. Aus diesen Anwendungen wurde der Bedarf nach einem effizienten Verfahren für die Zusammenführung verschlüsselter Identifikatoren deutlich. Über zahlreiche Zwischenstufen (Schnell/Bachteler/Reiher 2007) wurde von den Autoren dann das neue Verfahren entwickelt, implementiert und getestet (Schnell/Bachteler/Reiher 2009).

4 Technische Details des SAFELINK-Verfahrens

Das Problem der Verknüpfung von Datenbanken mit fehlerbehafteten Identifikatoren wie z. B. Namen lässt sich auf das Problem der Berechnung der Ähnlichkeit dieser Identifikatoren reduzieren. Da aus Gründen des Datenschutzes auf keinen Fall Identifikatoren übermittelt werden sollen, die einen Rückschluss auf Personen zulassen, müssen die Identifikatoren so verschlüsselt werden, dass sie anschließend nicht mehr entschlüsselt werden können. Das Problem besteht also in der Berechnung

1 Für das Verständnis des Verfahrens von Churches & Christen sind einige technische Begriffe, wie z. B. N-Gramme, erforderlich, die wir erst später in diesem Aufsatz einführen. Die Kenntnis dieser Begriffe vorausgesetzt lässt sich das Verfahren folgendermaßen zusammenfassen: Beide Datengeber bilden für jeden Namen die Potenzmenge seiner N-Gramm Menge und verschlüsseln jede Teilmenge der Potenzmenge gesondert durch einen HMAC Algorithmus. Die resultierenden Hashwerte werden zusammen mit der Zahl der in der Teilmenge enthaltenen N-Gramme an eine Drittpartei übermittelt. Für jedes Paar an Namen kann die Drittpartei anhand der größten übereinstimmenden Teilmenge der beiden Namen die N-Gramm Ähnlichkeit bestimmen. Für Einzelheiten muss auf die Publikation von Churches & Christen verwiesen werden. Eine Kritik der Effizienz des Verfahrens findet sich bei (Verykios/Karakasidis/Mitrogiannis 2009).

2 Literaturübersichten finden sich bei Trepetin (2008) und Schnell, Bachteler & Reiher (2009).

von Ähnlichkeiten zwischen unentschlüsselbaren Identifikatoren. Exakt dies ist die Aufgabenstellung für das von uns neu entwickelte Verfahren „SAFELINK". Um das Verfahren zu erläutern, ist die Einführung einiger technischer Begriffe erforderlich.

Viele Algorithmen der Informatik für Zeichenfolgen basieren auf N-Grammen. Unter einem N-Gramm versteht man eine Folge aus N Zeichen; üblich in diesem Kontext sind Bigramme (N=2) und Trigramme (N=3). Häufig wird eine Zeichenkette (z. B. „Reisebus") vor der Zerlegung in N-Gramme an beiden Enden mit N-1 Leerzeichen ergänzt, so dass sich im Beispiel die Bigramm-Menge {_R; RE; EI; IS; SE; EB; BU; US; S_} ergibt. Die Ähnlichkeit von zwei N-Gramm-Mengen wird häufig mit dem Dice-Koeffizienten

$$\frac{2c}{a+b} \tag{1}$$

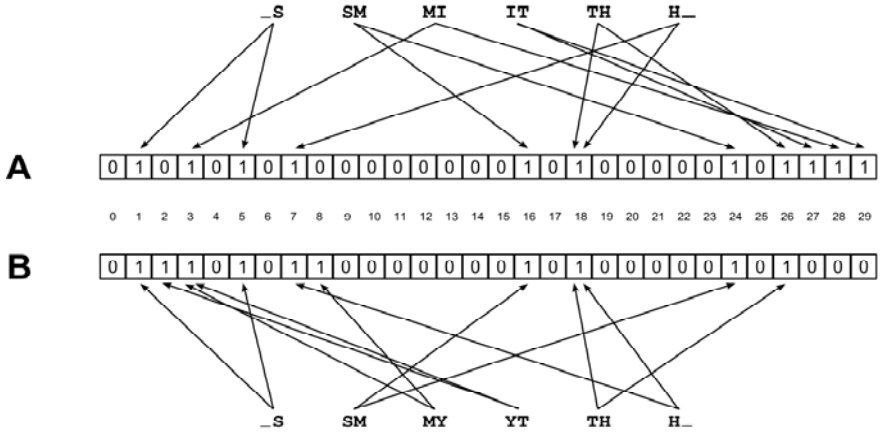
angegeben, wobei a die Zahl der N-Gramme in der Zeichenkette A, b die Zahl der N-Gramme in der Zeichenkette B und c die Zahl der N-Gramme ist, die in beiden Zeichenketten vorkommen. Viele Verfahren zur Zusammenführung von Datenbanken basieren darauf, dass diejenigen Records zusammengeführt werden, deren Identifikatoren sich am stärksten ähneln. Eine einfache Möglichkeit besteht in der Wahl derjenigen Paare, deren Dice-Koeffizienten ihrer N-Gramme am höchsten sind.

Ein Vektor der Länge l aus Nullen und Einsen wird in der Informatik als Bit-Vektor bezeichnet. Bildet man einen numerischen Wert n aus der Menge der natürlichen Zahlen mit einer Funktion auf das Intervall $0 \leq n \leq l$ ab, so wird die Funktion als „Hash-Funktion" bezeichnet. Die Abbildung mehrerer Hash-Funktionen auf einen Bit-Vektor ist ein sogenannter „Bloom-Filter" (Bloom 1970). Bloom-Filter werden meist zum Test der Mitgliedschaft eines Objekts in einer Menge verwendet (Brass 2008). Von besonderem Interesse ist die Verwendung von kryptografischen Hash-Funktionen. Hierbei handelt es sich um Einwegfunktionen, bei denen vom Ergebnis nicht mehr auf die Eingabewerte der Funktionen geschlossen werden kann. Das klassische Beispiel für eine solche Funktion ist der häufig für Prüfsummen eingesetzte Algorithmus MD-5, ein moderneres Beispiel SHA-1 (Swoboda/Spitz/Pramateftakis 2008).

Um die Ähnlichkeit zwischen Identifikatoren zu berechnen, ohne die Identifikatoren offenzulegen, speichern wir im Safelink-Verfahren die N-Gramme jedes Namens in einem eigenen Bloom-Filter. Für die Speicherung werden immer mehrere unabhängige Hash-Funktionen verwendet. Für die Zuordnung der Namen aus verschiedenen Datenbanken werden dann nicht mehr die Namen verglichen, sondern nur noch bitweise die Bloom-Filter.

Abbildung 2 illustriert das Verfahren für die beiden Namen SMITH und SMYTH mit Bigrammen, einem (unrealistisch kurzen) Bloom-Filter mit 30 Bits und zwei Hash-Funktionen. Die Namen werden in Bigramme zerlegt und jedes Bigramm der beiden Namen in den Bloom-Filtern A und B gespeichert. So erbringt z. B. das gemeinsame Bigramm „_S" den Hash-Funktionswert 1 für die erste Hash-Funktion und den Wert 5 für die zweite Hash-Funktion: Entsprechend werden die Bits an den Positionen 1 und 5 in beiden Bloom-Filtern auf den Wert 1 gesetzt. Im Gegensatz dazu sind die Bigramme „YT" (Hash-Werte 2 und 3) und „IT" (Hash-Werte 27 und 29) nur in einem Namen vorhanden, daher werden in den beiden Bloom-Filtern unterschiedliche Bits auf den Wert 1 gesetzt.

Abbildung 2 Beispiel für die Verwendung zweier Bloom-Filter für die Berechnung von Stringähnlichkeiten



Nach der Speicherung aller Bigramme in die Bloom-Filter sind 8 identische Bit-Positionen in beiden Bloom-Filtern auf 1 gesetzt. Insgesamt sind 11 Bits im Filter A und 10 Bits in Filter B gleich 1 gesetzt. Der Dice-Koeffizient ergibt sich als $\frac{2 \cdot 8}{(10+11)} \approx 0{,}762$. Die Ähnlichkeit der Namen kann also allein durch die Ähnlichkeit der Bloom-Filter approximiert werden. Da aber für die Speicherung kryptografische Einwegfunktionen genutzt wurden, können aus den Bloom-Filtern die Eingabenamen nicht mehr rekonstruiert werden. Dadurch wird eine fehlertolerante Verknüpfung zweier Datenbanken durch eine Forschungsgruppe oder eine Drittpartei bei vollständiger Anonymität der Identifikatoren möglich.

Für den Einsatz des Verfahrens sind eine Reihe von Entscheidungen notwendig. Weitgehend unkritisch ist die Wahl, ob Bi- oder Trigramme eingesetzt

werden: Unsere Simulationen zeigen kaum Unterschiede zwischen den Ergebnissen. Ebenso ist die Wahl der Länge der Bloom-Filter innerhalb eines Intervalls von 500-1.000 Bits eher unproblematisch. Derzeit neigen wir eher zu längeren Bloom-Filtern.[3] Die Wahl der eigentlichen Hash-Funktion erscheint uns auf der Grundlage der Arbeiten von Kirsch und Mitzenmacher (2006) ebenfalls weitgehend unproblematisch. Sie schlagen vor, mit zwei unabhängigen Hash-Funktionen k Hash-Funktionen zu realisieren. Die k Hash-Werte ergeben sich durch

$$g_i(x) = h_1(x) + ih(x) \bmod l \tag{2}$$

wobei i von 0 bis k-1 läuft und l die Länge des Bit-Arrays ist. Wir verwenden die erwähnten kryptografischen Funktionen SHA1 (h_1) und MD5 (h_2). Die Wahl der Zahl k der Hash-Funktionen beeinflusst das Verhalten des Verfahrens deutlich. Daher stellen wir dies weiter unten ausführlich dar.

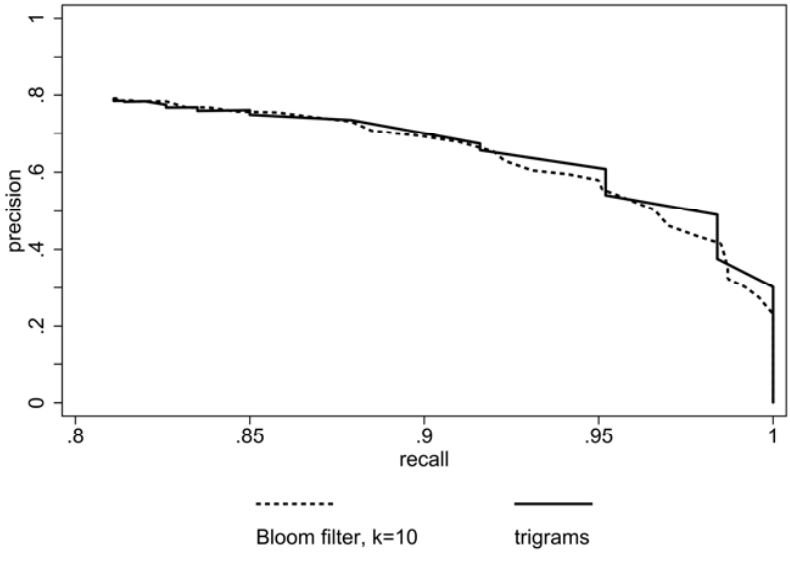
5 Empirische Tests des Verfahrens

Wir haben das Verfahren sowohl mit simulierten als auch mit echten Daten getestet. Um den Effekt der Anzahl der Hash-Funktionen auf die Leistung des Verfahrens zu testen, haben wir in einem Experiment mit simulierten Daten die Anzahl der Hash-Funktionen variiert (k = 1, 5, 10, 25, 50, 100). Die Länge der Filter war dabei konstant 1.000 Bits. Verglichen wurde mit den Ergebnissen unverschlüsselter Trigramme. Die Ausgangsdaten waren dabei 1.000 verschiedene aus einer Telefon-CD zufällig ausgewählte Nachnamen. Aus dem Datenbestand wurden alle nicht-alphabetischen Zeichen entfernt, die Umlaute umgewandelt und häufige Namensbestandteile wie akademische Titel und ehemalige Adelsprädikate gelöscht. Diese bereinigte Liste wurde kopiert und in der Kopie in jedem Namen mit der Wahrscheinlichkeit 0,25 an einer jeweils neu ermittelten zufälligen Stelle exakt ein Buchstabe durch einen anderen Buchstaben ersetzt. Auf diese Weise erhielten wir zwei Listen von jeweils 1.000 Namen, bei denen 250 Namen exakt eine Differenz aufwiesen.

3 Die Länge der Bloom-Filter beeinflusst auf recht komplizierte Weise die Sicherheit des Verfahrens: Längere Schlüssel sind für eine bestimmte Form des Angriffs (einen Wörterbuchangriff) weniger sicher als kurze. Der geringe Verlust an Sicherheit bei einem langen Schlüssel wird aber durch den Gewinn an Präzision deutlich aufgewogen. Eine genaue Analyse dieses Problems ist der Gegenstand einer laufenden Studie der Arbeitsgruppe.

Ein Beispiel für die Ergebnisse (hier mit 10 Hash-Funktionen) zeigt Abbildung 3 mit einem sogenannten „precision versus recall plot". In einem PR-Plot wird die Genauigkeit eines Abrufs aus einer Datenbank („precision") gegen die Empfindlichkeit des Abrufs („recall") geplottet. PR-Plots sind die in der Informatik übliche Variante der in der medizinischen Literatur gebräuchlichen Receiver-Operating-Characteristic-Kurven, bei denen Sensitivität eines Verfahrens gegen die Spezifität des Verfahrens abgebildet wird (Davis/Goadrich 2006).

Abbildung 3 Precision-Recall-Kurven von unverschlüsselten Trigrammen und Bloom-Filtern der Länge 1.000 mit 10 Hash-Funktionen



Exakter: Für ein gegebenes Maß an Ähnlichkeit zweier Schlüssel wird ein Record-Paar als „match" bezeichnet, wenn die Records tatsächlich zusammengehören. Alle anderen Paare sind daher „non matches" (Winkler 1995). Entsprechend ergibt sich dann die übliche Klassifikation in korrekt übereinstimmende Paare („true positive", TP), falsch positive Paare (FP), falsch negative Paare (FN) und korrekt nicht übereinstimmende Paare („true negative", TN). Die Vergleichskriterien ergeben sich dann als

$$\text{recall} = \frac{\sum TP}{\sum TP + \sum FN} \tag{3}$$

$$\text{precision} = \frac{\sum TP}{\sum TP + \sum FP} \tag{4}$$

In Abbildung 3 ist die Precision-Recall-Kurve für die unverschlüsselten Trigramme der Precision-Recall-Kurve für die Bloom-Filter sehr ähnlich. Wenn man nur wenige Hash-Funktionen in einem Bloom-Filter verwendet, dann ist die Wahrscheinlichkeit, dass verschiedene Trigramme auf identische Bit-Positionen abgebildet werden, sehr klein. Je mehr Hash-Funktionen verwendet werden, desto eher werden verschiedene Trigramme auf die gleiche Position abgebildet. Dies ist für die Leistung von SAFELINK von zentraler Bedeutung: Mit steigender Zahl der Hash-Funktionen k wird ein Wörterbuch-Angriff ohne Kenntnis der Parameter der Verschlüsselung immer schwieriger.[4] Andererseits wird durch ein Ansteigen der Zahl der Hash-Funktionen die Wahrscheinlichkeit einer falsch-positiven Identifikation eines Record-Paares immer höher. Es handelt sich also um eine Abwägung zwischen erhöhter Sicherheit für einen sehr unwahrscheinlichen (und natürlich illegalen) Angriff und Verringerung der Präzision der Verknüpfung. Die Zahl der Hash-Funktionen sollte daher nicht zu hoch gewählt werden.

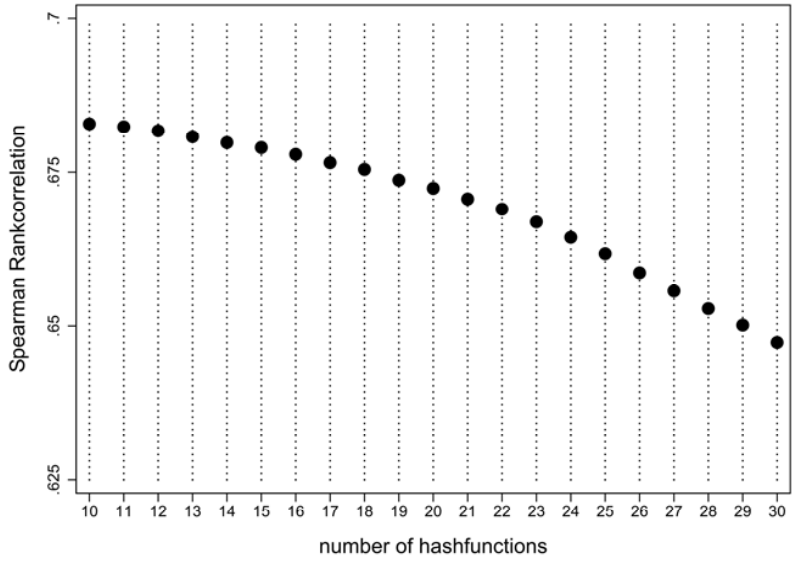
Nach unseren ersten Versuchen schien ein vernünftiges Intervall für die Zahl der Hash-Funktionen zwischen 10 und 30 zu liegen. Für dieses Intervall haben wir das Experiment für jede Zahl k zwischen 10 und 30 wiederholt. Für jedes Record in den beiden Datenbanken wurde die Ähnlichkeit anhand der unverschlüsselten Trigramme und der Bloom-Filter berechnet. Anschließend wurde die Rang-Korrelation zwischen diesen beiden Ähnlichkeiten berechnet. Das Ergebnis zeigt Abbildung 4.

Die Rangkorrelation zwischen den Ähnlichkeiten auf der Basis der unverschlüsselten Trigramme und der Bloom-Filter sinkt monoton mit steigender Zahl k der Hash-Funktionen. Bei 30 Hash-Funktionen liegt der Rangkorrelationskoeffizient nur noch bei 0,647. Bis weitere Ergebnisse auf der Basis realer Datensätze vorliegen, halten wir daher 15 Hash-Funktionen für einen akzeptablen Kompromiss.

In einem zweiten Test wurde SAFELINK mit den Ergebnissen unverschlüsselter Bigramme und der sogenannten „Kölner Phonetik" (Postel 1969) verglichen. Phonetiken sind Sammlungen von Regeln, die eine Zeichenkette in einen phonetischen Code übersetzen. Falls die phonetischen Codes übereinstimmen, wird dem Paar als Ähnlichkeitswert die Zahl 1 zugewiesen, sonst die Zahl 0.

4 In der Kryptografie bezeichnet man den Vergleich der verschlüsselten Werte mit den Verschlüsselungen bekannter Eingabewerte in den Verschlüsselungsalgorithmus als Wörterbuchangriff.

Abbildung 4 Rangkorrelationen zwischen den Dice-Koeffizienten von unverschlüsselten Trigrammen und Bloom-Filtern der Länge 1.000 nach Zahl der Hash-Funktionen



Die Kölner Phonetik wurde speziell für den deutschen Sprachraum entwickelt und wird häufig verwendet, so z. B. von den deutschen Krebsregistern (Eichelberg/Aden/Thoben 2005). Für diesen zweiten Test des Verfahrens wurden ausschließlich die Namen (ohne jede inhaltliche Information) aus zwei Verwaltungsdatenbanken mit jeweils ca. 15.000 Einträgen verwendet. Die Aufgabe bestand darin, zusammengehörende Namen in beiden Datenbanken zu finden. Bei diesem Test haben wir Bigramme, 15 Hash-Funktionen und Bloom-Filter mit 500 Bits verwendet. Die Leistung der unverschlüsselten Bigramme, der Kölner Phonetik und SAFELINK wurde dadurch verglichen, dass für jedes der drei Verfahren ein unabhängiges Record-Linkage mit exakt denselben Parametern durchgeführt wurde. Für das Record-Linkage wurde unser Programm „MTB" (Schnell/Bachteler/Reiher 2005) verwendet. Abbildung 5 zeigt den PR-Plot von SAFELINK im Vergleich zu den unverschlüsselten Bigrammen, Abbildung 6 im Vergleich zur Kölner Phonetik. SAFELINK erzielt nahezu die gleiche Leistung wie die unverschlüsselten Bigramme und bessere Ergebnisse als die Kölner Phonetik. Dies gilt vor allem für Recall-Level über 0,75, da die Phonetik dann besonders viele falsch-positive Ergebnisse erbringt.

Abbildung 5 Precision-Recall-Kurven von unverschlüsselten Bigrammen und Bloom-Filtern der Länge 500 mit 15 Hash-Funktionen

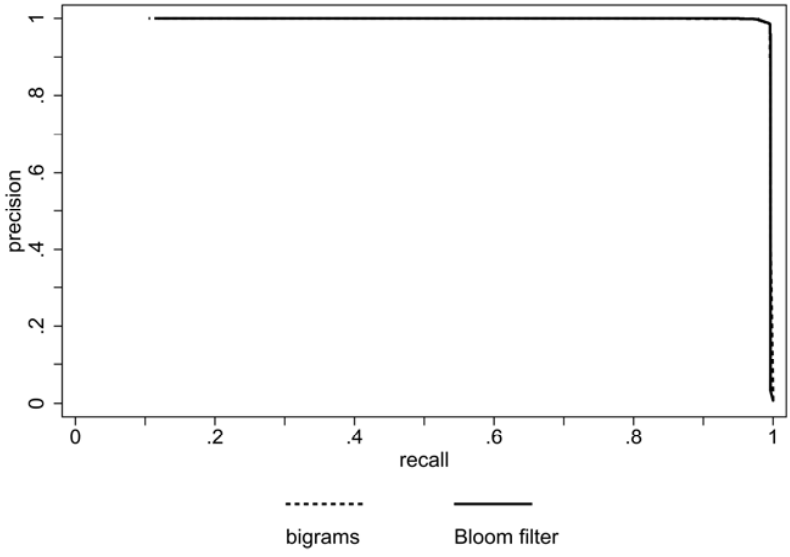
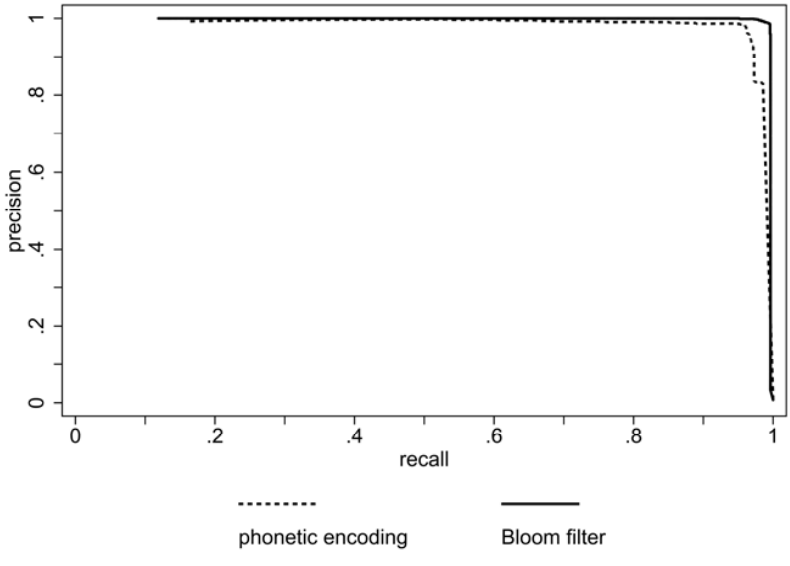


Abbildung 6 Precision-Recall-Kurven der Kölner Phonetik und Bloom-Filtern der Länge 500 mit 15 Hash-Funktionen



6 Ein Protokoll zur Datenverknüpfung mit SAFELINK

Die zuvor beschriebenen erfolgreichen Tests legen es nahe, die vorgeschlagene Methode innerhalb eines Protokolls für das datenschutzrechtlich unbedenkliche Record-Linkage zu verwenden. Um dem Protokoll noch eine weitere Sicherheitsschicht hinzuzufügen, schlagen wir vor, die Hash-Funktionen SHA1 und MD5 durch deren kryptografische Varianten mit einem Schlüssel K zu ersetzen.[5] Bei diesen sogenannten „keyed hash message authentication codes" (HMAC) wird ein zusätzlicher Schlüssel K verwendet. Bei einem HMAC ist es auch bei Kenntnis des Schlüssels K nicht möglich, aus einem gegebenen HMAC-Wert eine dazu passende Nachricht zu rekonstruieren (Swoboda/Spitz/Pramateftakis 2008: 108).

Basierend auf der so erweiterten Bloom-Filter-Methode ist die Implementation eines Record-Linkage-Protokolls recht einfach. Unser Protokoll verwendet eine halb vertrauenswürdige Drittpartei[6], da ansonsten einer der beider Datengeber A oder B versuchen könnte, mit Hilfe einer Liste möglicher Einträge die Bloom-Filter des anderen Datengebers zu identifizieren.[7] Neben den Datengebern A und B mit den Datenbeständen S_a und S_b werden die Drittpartei C und der Empfänger D der zusammengeführten Datensätze im Protokoll benötigt. Das Protokoll läuft folgendermaßen ab:

1. Die Datengeber A und B einigen sich über die Länge l der Bloom-Filter, die Anzahl k der Hash-Funktionen sowie über den vertraulichen Schlüssel K.
2. Für jeden Namen i in seinem Datenbestand führt jeder der Datengeber die folgenden Schritte durch:
   a) Umwandlung des Namens in seine N-Gramme.
   b) Speichern der N-Grammmenge mit k Funktionen und dem Schlüssel K in einem Bloom-Filter der Länge l.
   c) Speichern des Bloom-Filters und einer systemfreien, eindeutigen und zufällig generierten Identifikationsnummer id in eine Liste BF. Hinzufügen der Identifikationsnummer zum Datenbestand.

5 Diese Varianten werden als HMAC-MD5 und HMAC-SHA1 bezeichnet und wurden von Krawczyk, Bellare & Canetti (1997) vorgeschlagen.

6 In der englischsprachigen Literatur wird zwischen „trusted third parties" (TTPs) und „semi-trusted third parties" (STTPs) unterschieden. Für eine STTP gelten weniger anspruchsvolle Annahmen bezüglich ihrer Vertrauenswürdikeit als für eine TTP. Gefordert wird, dass eine STTP ein vereinbartes Protokoll einhält und nicht böswillig mit einer anderen Partei kooperiert. Im Gegensatz zu einer TTP wird aber nicht angenommen, dass eine STTP nicht versucht, einen kryptanalytischen Angriff auf verschlüsselte Informationen zu unternehmen. Für das Safelink-Verfahren ist die Annahme einer STTP ausreichend.

7 Ein solcher Wörterbuchangriff wäre möglich, falls die Datengeber die Zahl der Hash-Funktionen k, den Schlüssel K und die Länge des Bloom-Filters l kennen. Weiterhin wüssten die Datengeber welche Einträge in beiden Datenbanken vorhanden wären. Beide Probleme werden durch den Einsatz einer Drittpartei vermieden.

3. Beide Datengeber entfernen aus ihren Datenbeständen alle Namen und sonstige Identifikatoren bis auf id.
4. Beide Datengeber übersenden ihre Datenbestände (ohne Identifikatoren bis auf id) an D.
5. Beide Datengeber übersenden ihre Bloom-Filter-Listen samt id an die Drittpartei C.
6. C vergleicht alle Paare von Bloom-Filtern und berechnet die Ähnlichkeit der Filter mit dem Dice-Koeffizienten.
7. Paare mit den höchsten Dice-Koeffizienten werden zu Tupeln (id in BF_a, id in BF_b, Dice-Koeffizient) zusammengefasst.
8. C übersendet die Liste der besten Paare mit ihren Tupeln an D.
9. D führt die Datenbestände unter Verwendung der Tupel zusammen.

Das Protokoll ist in Hinsicht auf die Datengeber A und B sicher, da keiner der beiden Zugang zu den Bloom-Filtern des anderen hat. Die Drittpartei sieht nur die Bloom-Filter, kennt aber weder k, noch K, noch l. Selbst wenn er alle diese Informationen hätte, müsste die Drittpartei zusätzlich eine Liste potenzieller Namen abfragen. So lange die Drittpartei also nicht mit einem Datengeber konspiriert, ist das Protokoll sicher.[8]

7 Implementierung des Protokolls

Einer der Autoren hat das Verfahren zunächst innerhalb unseres Record-Linkage-Programms MTB in Java implementiert. Um potenzielle Fehler auszuschalten, hat ein anderer der Autoren vollkommen unabhängig davon das Protokoll erneut mit Python implementiert.[9] Beide Implementierungen umfassen ca. 100 Zeilen Programm-Code, wobei aber zahlreiche (jeweils andere) Standardbibliotheken eingebunden werden. Die Ergebnisse beider Programme sind identisch. MTB führt die Berechnungen dabei deutlich schneller aus, ca. 1.000 Records können dabei in 5 Minuten vollständig paarweise verglichen werden. Dies entspricht bei Anwendungen in der Praxis der maximalen Größe einer Teilmenge einer Datenbank, die tatsächlich vollständig

8 Sollte die Drittpartei mit einem Datengeber konspirieren, ist der direkte Austausch der Datenbestände ohnehin einfacher.

9 Da beide Sprachen plattformunabhängig sind, sollten die Programme unter jedem modernen Betriebssystem (Linux, Mac-OS, Windows) funktionieren.

verglichen wird.[10] Der Geschwindigkeitsverlust gegenüber der Verwendung unverschlüsselter Identifikatoren liegt bei ca. 20 %.

8 Zukünftige Weiterentwicklungen und Anwendungen

Das Kernproblem der fehlertoleranten Verknüpfung von stark kryptografierten Namen wurde von uns mit SAFELINK gelöst. Trotz der Einsatzbereitschaft der Programme und des Protokolls bleiben einige Probleme offen. Bei der Arbeit an diesem Protokoll wurde deutlich, dass die Art der Aufbereitung der Namen vor der Verschlüsselung noch erheblichen Forschungsbedarf besitzt. Weiterhin ist in der Literatur das Problem der kryptografischen Verschlüsselung numerischer Daten mit kardinalen Eigenschaften derart, dass Abstandsberechnungen zwischen den verschlüsselten Daten möglich bleiben, bislang nicht befriedigend gelöst. Dies ist z. B. für die Berücksichtigung des Geburtsdatums beim Record-Linkage von zentraler Bedeutung. Zusammen mit SAFELINK würden dadurch Personenidentifikatoren möglich, die dezentral immer neu generiert werden können, ohne dass sie zentral gespeichert werden müssen. Damit wären Verknüpfungen von personenbezogenen Daten im Längsschnitt möglich, die bisher aus rechtlichen Gründen kaum realisierbar waren. Neben den offensichtlichen Möglichkeiten von medizinischen Längsschnittstudien ergeben sich so z. B. technische Realisierungsmöglichkeiten für kriminologische Panels oder ein Bildungspanel mit Individualdaten im Längsschnitt. Die Lösung dieser Probleme ist Gegenstand unserer laufenden Bemühungen.

Literatur

Bethlehem, J., 2008: Surveys without questions. S. 500-511 in: E. D. de Leeuw, J. D. Hox und D. A. Dillman (Hg.): International handbook of survey methodology, New York: Erlbaum.

Bloom, B. H., 1970: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM 13: 422-426.

Brass, P., 2008: Advanced data structures. Cambridge: Cambridge University Press.

Churches, T. und P. Christen, 2004: Some methods for blindfolded record linkage. BMC Medical Informatics and Decision Making 4: 1-17.

Davis, J. und M. Goadrich, 2006: The relationship between precision-recall and ROC curves. S. 233-240 in: Proceedings of the 23rd International Conference on Machine Learning, New York.

10 Beim Record-Linkage vermeidet man aus Rechenzeitgründen den vollständigen Paarvergleich zweier Datenbanken. Bei großen Datenbeständen wird immer nur innerhalb von Teilmengen („blocks“) verglichen. Die Bildung dieser Blöcke ist ein eigenes Forschungsgebiet innerhalb des Record-Linkage.

Eichelberg, M., T. Aden und W. Thoben, 2005: A distributed patient identification protocol based on control numbers with semantic annotation. International Journal on Semantic Web and Information Systems 1: 24-43.

Herzog, T. N., F. J. Scheuren und W. E. Winkler, 2007: Data quality and record linkage techniques. New York/Berlin: Springer.

Kirsch, A. und M. Mitzenmacher, 2006: Less hashing, same performance: building a better Bloom filter. S. 456-467 in: Y. Azar und T. Erlebach (Hg.): Algorithms-ESA 2006. Proceedings of the 14th Annual European Symposium, 11-13 September 2006, Zürich. Berlin: Springer.

Krawczyk, H., M. Bellare und R. Canetti, 1997: HMAC: Keyed-hashing for message authentication. Internet RFC 2104. http://tools.ietf.org/html/rfc2104 (9.9.2009).

Postel, H. J., 1969: Die Kölner Phonetik. Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse. IBM-Nachrichten 19: 925-931.

Schnell, R. und T. Bachteler, 2006: Der Bedarf nach einer Treuhänderlösung für die Verknüpfung von Mikrodaten in der Bundesrepublik. Diskussionspapier, Zentrum für Quantitative Methoden und Surveyforschung, Universität Konstanz. http://www.uni-due.de/methods/documents/SchnellDatenTreuhandRatWSD.pdf (9.9.2009).

Schnell, R., T. Bachteler und J. Reiher, 2009: Privacy-preserving record linkage using Bloom filters. BMC Medical Informatics and Decision Making 9: 41.

Schnell, R., T. Bachteler und J. Reiher, 2007: Die sichere Berechnung von Stringähnlichkeiten mit Bloom-Filtern, Diskussionspapier, Universität Konstanz, September 2007.

Schnell, R., T. Bachteler und S. Bender, 2004: A toolbox for record linkage. Austrian Journal of Statistics 33: 125-133.

Schnell, R., T. Bachteler und J. Reiher, 2005: MTB: Ein Record-Linkage-Programm für die empirische Sozialforschung. Zentralarchiv-Informationen 56: 93-103. (http://www.za.uni-koeln.de/publications/pdf/za_info/ZA-Info-56.pdf (9.9.2009).

Schnell, R., P. Hill und E. Esser, 2008: Methoden der empirischen Sozialforschung. 8. Auflage, München: Oldenbourg.

Schnell, R., 2009: Record linkage from a technical point of view, Expertise für den Rat für Wirtschafts- und Sozialdaten, Projekt: Developing the Research Infrastructure for the Social and Behavioral Sciences in Germany and Beyond: Progress since 2001, Current Situation and Future Demands, Februar 2009.

Swoboda, J., S. Spitz und M. Pramateftakis, 2008: Kryptographie und IT-Sicherheit: Grundlagen und Anwendungen. Wiesbaden: Vieweg+Teubner.

Trepetin, S., 2008: Privacy-preserving string comparisons in record linkage systems: a review. Information Security Journal: A Global Perspective 17: 253-266.

Verykios, V. S, A. Karakasidis und V. K. Mitrogiannis, 2009: Privacy preserving record linkage approaches. International Journal of Data Mining, Modelling and Management 1: 206-221.

Winkler, W. E., 1995: Matching and record linkage. S. 355-384 in: B. G. Cox, D. A. Binder, B. N. Chinnappa, A. Christianson, M. J. Colledge und P. S. Kott (Hg.): Business survey methods. New York: Wiley.

Anschrift der Autoren

Prof. Dr. Rainer Schnell
Tobias Bachteler
Jörg Reiher
Universität Duisburg-Essen
Lotharstraße 65
47057 Duisburg
rainer.schnell@uni-due.de
www.methodenzentrum.de