Schlüsselaustausch und Policy Enforcement bei zweckgebundener Datenübermittlung

Kai Wagner

Wirtschaftswissenschaftliche Fakultät
Universität Regensburg
93040 Regensburg
kai.wagner@gmx.de

Abstract: Im beschriebenen Szenario überlässt ein Betroffener verschiedenen Datenverarbeitern unter dem Gebot strikt zweckgebundener Nutzung eine anlassbezogene Auswahl seiner Daten. Zum Zeitpunkt der Datenhinterlegung und Policy-Formulierung kennen sich Betroffener und potentieller Verarbeiter nicht. Sie nutzen zwei weitere Parteien als Vermittlungsdienst, die über die Einhaltung der Policies wachen, ohne sie selbst interpretieren zu können. Es wird gezeigt, wie die Vermittler ihre Rolle wahrnehmen, und dabei nahezu keine Informationen über den Datenaustausch erhalten. Die Realisierung des Policy-Abgleichs nutzt einen One-Time-Pad-basierten asynchronen Schlüsselaustausch zwischen Betroffenem und Verarbeiter, in den die Vermittler einbezogen werden, wiederum ohne zusätzliche Kenntnisse über Art und Inhalt der Kommunikation zu erwerben.

1 Einführung

Das betrachtete Anwendungsszenario basiert auf dem Bedarf eines Dateneigners[1], auf seine Person bezogene Datensätze für mögliche Verarbeiter bereit zu halten. Dabei liegt das Augenmerk darauf, die Daten einem Verarbeiter nur im erforderlichen Umfang und erst dann zugänglich zu machen, wenn dieser einen vorgesehenen und aktuell vorliegenden Anlass zur Datenverarbeitung vorweist. Da die weitere Verwendung einmal aufgedeckter Daten nur schwer zu kontrollieren ist, kommt dem spätest möglichen Zeitpunkt ihrer Bekanntgabe besondere Bedeutung zu. Gleichzeitig soll der Datenverarbeiter darauf vertrauen können, dass im Bedarfsfall die benötigten Daten vorliegen, unabhängig davon, ob der Dateneigner in diesem Moment erreichbar ist. Zusätzlich ist natürlich gefordert, dass die Datenübermittlung zwischen den Parteien vertraulich stattfindet, nicht geleugnet und nicht unerkannt verfälscht werden kann. Ein Protokoll, das die genannten Aspekte erfüllt, kann unter anderem in den folgenden Praxisszenarien Anwendung finden:

[1] In Bezug auf personenbezogene Daten wird im Folgenden der Dateneigner dem Betroffenen gleich gesetzt.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

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

Komplexer wird die Situation, wenn ein im Notfall erstversorgender Arzt nicht wie der Betriebsarzt vorab bekannt ist, sondern zunächst nur über die Zugehörigkeit zur Gruppe „Notfall-Ärzte“ identifiziert wird. Damit die geeigneten Daten zur Verfügung stehen, muss der Eigner vorab festlegen, welcher Verarbeitergruppe Daten zugänglich gemacht werden sollen, ohne die Gruppenmitglieder individuell benennen zu können.

Aus dem nicht-medizinischen Bereich: Auf einem Online Marktplatz bzw. einer Auktionsplattform für elektronische Güter treten Käufer nur mit ihren Pseudonymen auf. Kommt es zum Abschluss, werden die finanzielle Transaktion und die Übermittlung der Inhalte durch einen Treuhänder-Service des Plattform-Betreibers abgewickelt, wobei Käufer und Verkäufer voreinander anonym bleiben. Kommt es jedoch zu einer Beschwerde seitens des Käufers, weil die Inhalte nicht den zugesagten Eigenschaften entsprechen, erwartet er eine Garantie, dass er die echten Kontaktdaten des Verkäufers in Erfahrung bringen kann.[2]

Ähnlich gelagerte Szenarien finden sich etwa bei Reisen, deren genaues Ziel nur im Notfall bekannt werden soll, oder bei der Auswertung von Daten aus einem Personalverwaltungssystem, die nur bei verdachtsbezogenem Systemaudit einer Gruppe von Prüfern zugänglich zu machen sind. Allen Szenarien sind die Grundzüge gemeinsam: Daten werden abgelegt, die nur streng zweckgebunden an Verarbeiter herausgegeben werden, wobei sich die Identität der Verarbeiter auch erst nach der erfolgten Datenablage herausstellen kann. In diesem Sinne geht die Menge der denkbaren Szenarien durchaus auch über den Fokus auf personenbezogene Daten hinaus.

Das im Folgenden vorgestellte Protokoll kombiniert bestehende kryptographische Verfahren um die beschriebenen Anforderungen zu erfüllen. Ein Proof-of-Concept befindet sich in der Implementierung am Fachbereich Informatik der Universität Hamburg.

[2] Siehe dazu auch den „Identitäts-Treuhänder“ bei [FP02].

2 Datenablage und Policy-Formulierung

Der Dateneigner A überlässt seine Daten in verschlüsselter Form einem Treuhänder-Service T, bei dem sie von Mitgliedern der Verarbeitergruppe B abgerufen werden können, sobald der festgelegte Anlass eintritt. Sei Y_l die Definition eines Datentyps aus einem allen Parteien bekannten Vokabular mit $l \in \{1, 2, \ldots, p\}$ und p = Anzahl der vereinbarten Datensatztypen, dann ist $M(A, Y_l) = M_{Al}$ ein Datensatz des Typs Y_l von und über den Betroffenen A. Die Datensätze werden durch A mit einer „Sticky Policy" versehen, die ihre Herausgabe regelt.[3]

Sei J_k ein möglicher Zweck der Datenverarbeitung aus einem allen Parteien bekannten Vokabular mit $k \in \{1, 2, \ldots, o\}$ und o = Anzahl der vereinbarten Verwendungszwecke[4]. Dann formuliert P_{ABlk} eine Policy, die festlegt, dass ein Mitglied der Gruppe B den Datensatz des Betroffenen A vom Typ Y_l nur dann abrufen darf, wenn er dies mit dem Anlass J_k begründet. Dabei ist die Policy keine schlichte Reihung ihrer 4 Teilaspekte, sondern:

$$P_{ABlk} = E_{AB}(Y_l, J_k),$$

wobei E_{AB} für die Verschlüsselung mit einem symmetrischen Schlüssel K_{AB} steht, der ein gemeinsames Geheimnis zwischen dem Betroffenen A und dem Mitglied der Empfängergruppe B ist. Für den Einsatz als E_{AB} eignen sich Blockchiffrierer wie 3DES/TDEA[5] oder AES/Rijndael[6].

In welcher Form der Dateneigner den von ihm erzeugten Schlüssel K_{AB} dem Empfängergruppen-Mitglied zukommen lässt, das er ja zum Zeitpunkt der Policy-Ablage nicht kennen muss, wird in Abschnitt 4 beschrieben. Hier ist zunächst davon auszugehen, dass der Schlüssel bereits den beiden Parteien, aber niemandem sonst, bekannt ist.

[3] Vgl. [APS02], [Mö06] zu den Charakteristika von Sticky Policies. Die Policies „kleben" an den Daten, für die sie formuliert wurden. Dazu werden sie häufig gemeinsam mit diesen in einem kryptographischen Container aufbewahrt, oder die Policies bilden einen Teil des Schlüssels zum Öffnen desselben.

[4] Dabei kann das Vokabular einem bereits etablierten System entnommen werden, solange dieses dem Dateneigner die geeigneten Bausteine an die Hand gibt, seine Datenschutz-Präferenzen zu formulieren [Ku07]. Beispielsweise ist die Eignung von P3P zu diskutieren, vgl. [Gr01], [Al07].

[5] NIST Special Publication 800-67 "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher" (http://csrc.nist.gov/publications/nistpubs/800-67/SP800-67.pdf).

[6] Federal Information Processing Standards Publication 197 "Advanced Encryption Standard", (http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf).

Auch die Datensätze M_{Al} werden mit K_{AB} verschlüsselt. So setzt sich jedes Datum, das der Betroffene in die Hände des Treuhänder-Service legt, aus dem Chiffrat des Datensatzes $E_{AB}(M_{Al})$ und dem Policy-Chiffrat $E_{AB}(Y_l, J_k) = P_{ABlk}$ zusammen. Da der Treuhänder den Schlüssel K_{AB} nicht kennt, kann er weder die Policy noch den Datensatz selbst lesen. Gelingt es dem Dateneigner zudem, seine Daten so über ein Anonymisierungsnetzwerk (z.B. AN.ON[7] oder JonDo[8]) zu übermitteln, dass er als Absender nicht identifizierbar ist, kann er gegenüber dem Treuhänder sogar vollständig anonym bleiben. Dem gegenüber hat der Verarbeiter ein legitimes Interesse, die Identität des Eigners zuverlässig feststellen zu können. Daher signiert der Dateneigner den Datensatz M_{Al} vor der Verschlüsselung digital, es entsteht also $E_{AB}(S_A(M_{Al}))$. Der Treuhänder verfügt im Gegensatz zum künftigen Verarbeiter nicht über den Schlüssel K_{AB}, kann in Konsequenz die äußere Verschlüsselung nicht entfernen und damit auch die Signatur S_A nicht testen.

Die (gegenüber dem Treuhänder) anonyme Datenablage birgt eine zusätzliche Herausforderung: Möchte der Dateneigner zu einem Zeitpunkt nach der initialen Ablage einen Datensatz beim Treuhänder ändern oder entfernen, muss er nachweisen können, dass er der legitime Eigner ist – wiederum ohne seine Identität preiszugeben. Realisiert wird dies dadurch, dass A bei der ersten Erstellung zu jedem Datensatz eine von ihm erzeugte Zufallszahl r_{Ax} an den Treuhänder übermittelt. Schickt er später eine Aktualisierungsaufforderung für einen Datensatz, weist er durch Kenntnis der Zufallszahl gegenüber dem Treuhänder nach, dass er für diesen Datensatz legitimiert ist. Um dabei Replay-Attacken [Sc09], [Sc96] durch mögliche Störer zu vermeiden, wird nicht direkt die Zufallszahl übermittelt, sondern ihr Hashwert[9]. Zur Validierung führt der Treuhänder die gleiche Hashfunktion durch. Bei Übereinstimmung des Ergebnisses mit dem übermittelten Wert gilt die Aktualisierung als legitim. Beide Parteien ersetzen im Anschluss in ihrer Ablage die Zufallszahl durch den eben verwendeten Hashwert, so dass ein erneutes Übermitteln einer abgefangenen alten Aktualisierungsanforderung durch einen Dritten nicht zum Erfolg führt.

Das an den Treuhänder übermittelte Datenpaket ist beispielhaft in Abbildung 1 dargestellt. Man beachte, dass die Datensätze für den Treuhänder keinen Informationsgehalt haben, da er nicht über den Schlüssel K_{AB} verfügt.

[7] http://www.anon-online.de
[8] http://www.anonym-surfen.de/jondo.html
[9] Dies geschieht durch Anwendung einer kryptographischen Einwegfunktion, die sich insbesondere durch eine hohe Schwierigkeit auszeichnet, aus dem Hashwert den Ausgangswert zu berechnen oder einen zweiten Ausgangswert mit identischem Hashwert zu ermitteln. Vgl. zur Definition geeigneter Algorithmen u.a. Federal Information Processing Standards Publication 180-3 „Secure Hash Standard“ (http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf).

ISBN 978-3-88579-286-4

Daten verschlüsselt	Policy verschlüsselt	Zufallszahl
$E_{AB}(S_A(M_A))$	$E_{AB}(Y_l , J_k)) = P_{ABlk}$	r_{Ax}
$E_{AB}(S_A$("AB negativ"))	E_{AB}("Blutgruppe", "med. Notfall")	345.456.685.461.239
$E_{AB}(S_A$("Havanna"))	E_{AB}("Urlaubsort", "Reisewarnung")	793.254.955.652.356
$E_{AB}(S_A$("Justus Mustermann"))	E_{AB}("Real-Name", "Klageerhebung")	773.432.982.345.615

Abbildung 1: Datenpaket beim Treuhänder

3 Durchsetzung der Policies

Für den künftigen Verarbeiter sind zwei verschiedene Abfragearten von Bedeutung. Zum einen besteht die Option, sich kundig zu machen, ob ein bestimmter Datensatz für ihn zur Verfügung steht. Damit kann er prüfen, ob ein Dateneigner seinen Verpflichtungen zur Datenablage nachgekommen ist, ohne dass er auf die Daten selbst zugreift. Diese Abfrage wird im Folgenden Existenzabfrage genannt. Zum anderen möchte er bei Vorliegen des entsprechenden Anlasses die Daten selbst abrufen. Dazu stellt er eine Inhaltsabfrage. Für diese liefert der Treuhänder den vollständigen Datensatz (Spalte „Daten verschlüsselt“ in Abbildung 1), während er im Fall einer Existenzabfrage nur das Vorliegen eines passenden Datensatzes bestätigt oder verneint.

Um valide Existenz- oder Inhaltsabfragen zu stellen, benötigt der Verarbeiter einen Identifikator, über den er mit dem Treuänder eindeutig kommunizieren kann. Im vorgestellten Protokoll stellt die Policy, die bei der Datenablage erstellt wurde, diesen Identifikator dar. Wie geschildert besteht eine Policy aus dem Tupel aus Datentyp und Verwendungszweck, sowie dem gemeinsamen Geheimnis von Dateneigner und Verarbeiter. Unter der Prämisse, dass der Datenverarbeiter über den Schlüssel K_{AB} verfügt, kann er eine Policy formulieren, die zu der an die gewünschten Daten gebundenen Policy identisch ist.

Die vom Verarbeiter B erzeugte Vergleichs-Policy

$$P_{ABlk}' = E_{AB}(Y_l, J_k)$$

wird zusammen mit der Kennzeichnung, ob es sich um eine Existenz- oder Inhaltsabfrage handelt, an den Treuhänder übermittelt. Dabei ist die Abfrage durch den Datenverarbeiter digital zu signieren. Zwar ist die Feststellung seiner Identität nicht unbedingt notwendig, um eine Herausgabe von Daten an unberechtigte Dritte zu vermeiden – schließlich verfügt nur der korrekte Verarbeiter über den Schlüssel K_{AB}, der ihm das Lesen der Daten erlaubt. Für die spätere Beweisbarkeit einer stattgefundenen Abfrage aus den Protokollen des Treuhänders ist sie aber von zentraler Bedeutung.

Erhält der Treuhänder eine Abfrage, testet er die Signatur des Verarbeiters, extrahiert die übersandte Vergleichspolicy und durchsucht seinen Datenbestand nach einem Satz, für den gilt:

$$P_{ABlk} = P_{ABlk}´$$

Findet er einen entsprechenden Satz antwortet er im Fall einer Existenzabfrage mit einer Bestätigung, im Fall einer Inhaltsabfrage mit dem verschlüsselten Datensatz $E_{AB}(S_A(M_{Al}))$. In beiden Fällen übernimmt er die Abfrage mitsamt der Signatur des anfragenden Verarbeiters in seine interne Protokolldatei. Die Abfragen ohne die zugehörige Signatur stellt er daneben öffentlich zur Verfügung (s. Abschnitt 5).

Erneut sei darauf verwiesen, dass der Treuhänder im Verlauf der geschilderten Schritte nur erfährt, welche Verarbeiter Anfragen stellen und ob diese erfolgreich sind. Er kann nicht feststellen, wessen Daten sie abfragen, welche Policies formuliert wurden oder wer die Dateneigner sind. Der von ihm durchgeführte Policy-Vergleich ist eine Vergleichsoperation auf Bit-Ebene, die semantische Ebene bleibt ihm verborgen.

4 Schlüsselaustausch

Die bisher beschriebenen Bestandteile des Protokolls basierten auf der Prämisse, dass Dateneigner und Verarbeiter über ein gemeinsames Geheimnis, den Schlüssel K_{AB}, verfügen. Nur unter dieser Voraussetzung können sie korrespondierende Policy-Chiffrate erzeugen, auf denen die Durchsetzung der Nutzungsregeln basiert. In einem einfachen Anwendungsszenario, bei dem sich Eigner und Verarbeiter vorab kennen, ist die Lösung trivial: Der Eigner erzeugt den Schlüssel K_{AB}, chiffriert ihn jeweils mit den öffentlichen Schlüsseln der vorgesehenen Verarbeiter und schickt ihn entweder direkt an diese oder hinterlegt ihn beim Treuhänder. Von dort können die Verarbeiter sich die Schlüssel jederzeit besorgen und zur Formulierung der Vergleichs-Policies nutzen.

Im hier beschriebenen Szenario kennt der Dateneigner die künftigen Verarbeiter jedoch nicht. Seine Daten- und Policy-Ablage erfolgt auf der Basis von Gruppen[10], deren Mitgliederlisten stetigen Veränderungen unterworfen sein können und deren Zusammensetzung nicht für den Zeitpunkt vorher gesagt werden kann, zu dem der festgelegte Anlass zur Datennutzung eintritt.

[10] Vgl. die eingangs beschriebenen Beispiele der Gruppen „Notfall-Ärzte“ oder „Systemauditoren“.

Zunächst wird eine unabhängige Instanz benötigt, die für die stetige Pflege der Gruppenzugehörigkeiten Sorge trägt. Diese Instanz zertifiziert neue Mitglieder der Gruppen, basierend auf den von diesen zu erbringenden Nachweisen, dass ihr Antrag auf Gruppenzugehörigkeit valide ist. Daher heißt die prüfende Instanz „Zertifizierer". Er darf nicht identisch mit dem Treuhänder sein, und beide dürfen keine böswillige Kooperation eingehen. Der Zertifizierer ist im Idealfall eine von der Öffentlichkeit kontrollierte Non-Profit-Organisation. Datenverarbeiter bewerben sich also beim Zertifizierer darum, in bestimmte Empfängergruppen aufgenommen zu werden. Sie legen die Nachweise vor, die der Zertifizierer prüft. Nach erfolgter Aufnahme in eine Gruppe aktualisiert der Zertifizierer die veröffentlichten Mitgliedslisten.

Der Zertifizierer übt im vorgestellten System eine weitere Funktion aus: Er übernimmt eine Vermittler-Rolle im Schlüsselaustausch. Der Dateneigner A erstellt zum Zeitpunkt der Datenablage für jede relevante und beim Zertifizierer Z bekannte Empfängergruppe B ein One-Time-Pad OTP_{AZ}, das mindestens die selbe Länge hat wie der signierte symmetrische Schlüssel K_{AB}[11]. A verschlüsselt die erstellten One-Time-Pads mit dem öffentlichen Schlüssel des Zertifizierers und sendet diesem die Liste der entstandenen $E_Z(OTP_{AZ})$. Z öffnet die Sendung durch Anwendung seines privaten Schlüssels und legt OTP_{AZ} in seiner Schlüsselverwaltung ab[12]. Gemeinsam mit der Kennzeichnung, für welche Gruppe ein OTP vorgesehen ist, wird es wie in Abbildung 2 gespeichert[13].

Verarbeitergruppe	One-Time-Pad verschlüsselt
B	**$E_Z(OTP_{AZ})$**

Abbildung 2: Ablage der One-Time-Pads beim Zertifizierer

Das OTP_{AZ} verwendet der Dateneigner zudem dazu, um den zuvor signierten Datenschlüssel K_{AB} mittels bitweisem XOR zu verschlüsseln. Versehen mit der Kennzeichnung der zugehörigen Verarbeitergruppe sendet der Eigner den Schlüssel-Satz, wie in Abbildung 3 zu sehen, an den Treuhänder. $OTP_{AZ}(x)$ bedeutet dabei die Anwendung des One-Time-Pad, das A für die Kommunikation mit Empfängergruppe B an Z geschickt hat, auf die Nachricht x.

[11] Das One-Time-Pad (auch „Vernam-Chiffre") ist ein Kryptosystem, das auf polyalphabetischer Substitution beruht und sich dadurch auszeichnet, dass der verwendete Schlüssel mindestens so lang ist wie der zu verschlüsselnde Klartext, vgl. [Sc09]. In der vorliegenden Arbeit bedeutet die Anwendung eines One-Time-Pad die bitweise Verknüpfung des Klartextes (hier: des Datenschlüssels) per Exklusiv-Oder (XOR) mit einer zufälligen Bitfolge von mindestens derselben Länge.

[12] Natürlich bedürfen die Verarbeitergruppen, individuellen Verarbeiter und Dateneigner zusätzlicher Indizierung in der Notation, da sie jeweils im Verhältnis n:m kombiniert sind. Darauf wird hier verzichtet, um die grundsätzliche Funktionsweise des Protokolls vereinfacht darzustellen.

[13] Zur Relevanz des klassischen One-Time-Pad für moderne Verfahren vgl. [Ri10a].

ISBN 978-3-88579-286-4

Verarbeitergruppe	Datenschlüssel verschlüsselt
B	**$OTP_{AZ}(S_A(K_{AB}))$**

Abbildung 3: Ablage des Datenschlüssels beim Treuhänder

Der Treuhänder kann den Datenschlüssel K_{AB} nicht aufdecken. Ihm fehlt das OTP_{AZ}, das er für die Entschlüsselung benötigen würde. Hier wird deutlich, dass eine böswillige Kooperation von Treuhänder und Zertifizierer das System korrumpieren würde, denn Z kennt das OTP_{AZ}.

Der Datenschlüssel liegt also beim Treuhänder, das zu seiner Entschlüsselung notwendige OTP beim Zertifizierer. Sobald ein Datenverarbeiter einer der definierten Gruppen beitritt, benötigt er die für die Gruppe festgelegten Datenschlüssel. Diese ruft er beim Treuhänder ab. Der Treuhänder prüft anhand der vom Zertifizierer veröffentlichten Listen die korrekte Zugehörigkeit zur Gruppe, identifiziert dann die bei ihm hinterlegten Schlüsselpakete zu dieser Gruppe und initiiert die Übermittlung der Schlüssel an das Empfängergruppen-Mitglied.

Dazu erstellt der Treuhänder ein weiteres One-Time-Pad, OTP_{TB}, und übermittelt es an den Datenverarbeiter. Gleichzeitig wendet er OTP_{TB} auf den verschlüsselten Datenschlüssel an, es entsteht $OTP_{TB}(OTP_{AZ}(S_A(K_{AB})))$. Dieses Chiffrat schickt er an den Zertifizierer, der darauf nun das von A hinterlegte OTP_{AZ} anwenden kann. Da die Anwendung von OTPs, also das Durchführen von bitweisem XOR, sowohl symmetrisch als auch kommutativ ist, bedeutet das:

$$OTP_{AZ}(OTP_{TB}(OTP_{AZ}(S_A(K_{AB})))) = OTP_{TB}(S_A(K_{AB}))$$

Dieses Chiffrat $OTP_{TB}(S_A(K_{AB}))$ sendet der Zertifizierer an den Datenverarbeiter, der nun seinerseits das zuvor vom Treuhänder erhaltene OTP_{TB} anwenden kann. Er prüft anschließend die Signatur des Dateneigners, legt den erhaltenen Datenschlüssel in seinem Schlüsselspeicher ab und kann mit dessen Hilfe im Bedarfsfall die Vergleichspolicies für Existenz- und Inhaltsabfragen erstellen.

Der jeweilige Kenntnisstand jeder teilnehmenden Partei während des Schlüsselaustauschs ist Abbildung 4 zu entnehmen.

Dateneigner	Aufbewahrer	Zertifizierer	Empfänger
OTP_{AZ}	→	OTP_{AZ}	
	OTP_{TB}	→	OTP_{TB}
K_{AB}	→ $OTP_{AZ}(S_A(K_{AB}))$	→ $OTP_{TB}(S_A(K_{AB}))$	→ K_{AB}

Abbildung 4: Kenntnisstand der Parteien beim Schlüsselaustausch

Alternative Protokolle zum Schlüsselaustausch sind zu prüfen: Diffie-Hellmann ist nicht verwendbar, da es sich um ein interaktives Verfahren handelt. Es erfordert, dass Dateneigner und Verarbeiter zu einem Zeitpunkt in eine direkte Kommunikation eintreten, was unter den formulierten Rahmenbedingungen nicht möglich ist. RSA als asymmetrischer Träger scheidet aus, da es – sofern es sicher implementiert ist – kein kommutativer Algorithmus ist[14]. Dies gilt auch für die meisten symmetrischen Blockchiffrierer der modernen Kryptographie. Das One-Time-Pad scheint also eine geeignete Wahl, zumal der Datenschlüssel, auf den es angewandt wird, verhältnismäßig kurz und von stabiler Länge ist[15].

One Time Pads können nur dann Sicherheit bieten, wenn ein zuverlässiger Zufallszahlengenerator zur Generierung der Pads zur Verfügung steht. Theoretisch sichere Generatoren sind mit relativ hohem Aufwand zu implementieren. Insbesondere sind sie vom Einsatz spezifischer Hardware abhängig. Die theoretisch perfekte Sicherheit des One Time Pad[16] ist aber ohnehin nicht auf das vorgestellte Verfahren zu übertragen, da die Pads selbst durch asymmetrische Kryptographie und die verschlüsselten Daten mit anderen symmetrischen Verfahren verschlüsselt sind, die zumindest gegenüber Brute-Force Angriffen keine absolute Sicherheit bieten. Die Zufallszahlen der One Time Pads absolut zufällig und den Einsatz im Verfahren damit theoretisch sicher zu machen, ist verzichtbar. Es genügt, wenn die erzeugten Zufallszahlen praktisch sicher sind, solange diese Sicherheit zu kompromittieren mindestens ebenso aufwändig ist, wie das Brechen der anderen eingesetzten Kryptosysteme[17].

5 Nachweis der Aktivitäten

T veröffentlicht ein jeweils aktuelles Log aller Anfragen auf seiner Website. Es beinhaltet jeweils die Angabe, ob es sich um eine Existenz- oder Inhaltsabfrage gehandelt hat, den Zeitstempel und die verschlüsselte Policy P_{ABlk}. Jeder Dateneigner A kann hier das Protokoll der für seine Datensätze gestellten Existenz- und Inhaltsabfragen anonym abrufen und so erfahren, ob ein Verarbeiter bereits Kenntnis von seinen Daten oder deren Existenz besitzt. Er wählt aus den von ihm initial erstellten verschlüsselten Policies P_{ABlk} diejenigen aus, für die er die Anfrageprotokolle einsehen möchte und durchsucht das Log nach diesen Policies. Eine formulierte Policy besitzt nur für den Dateneigner und den Verarbeiter Aussagekraft, da nur diese beiden Parteien den

[14] Vgl. [Si83], [De84].
[15] Von zentraler Bedeutung ist, dass das One-Time-Pad nicht zur originären Sicherung der Kommunikation verwendet wird. Es ersetzt die Sicherung der Kanäle, etwa durch Public-Key-Kryptographie, nicht. Würden alle beschriebenen Nachrichten ohne weiteren Schutz nur mit ihrer OTP-Kodierung verschickt, könnte ein Angreifer, der alle Nachrichten abfängt, diese mit XOR verknüpfen. Die verschiedenen OTPs der Nachrichten würden sich wegen der Kommutativität eliminieren, und es bliebe der signierte Datenschlüssel im Klartext (vgl. [Sc96], S. 591f).
[16] Der Beweis für die Sicherheit des OTP bei der Nutzung echter Zufallszahlen wird beispielsweise in [TW06] geführt: Zur Erfüllung von Shannons Forderung an ein perfekt sicheres Kryptosystem (Ein Angreifer erhält aus dem Kryptogramm keine neue Information) wird nachgewiesen, dass für eine gegebenes Kryptogramm, das mit einem echten OTP verschlüsselt wurde, alle möglichen Ausgangsnachrichten gleich wahrscheinlich sind.
[17] Vgl. [Ri10b], ebenso zu Optionen praktisch sicherer Zufallszahlenerzeugung.

ISBN 978-3-88579-286-4

Datenschlüssel zur Erzeugung des Policy-Chiffrats kennen. Niemand sonst kann aus den Log-Einträgen etwas über die kommunizierenden Parteien, den Inhalt oder Zweck der Datensätze erfahren.

Hält A bei der Analyse der Log-Einträge eine Anfrage für nicht gerechtfertigt, kann er entsprechende (juristische) Schritte gegen den Verarbeiter einleiten. Da die Veröffentlichung ohne die Signatur des Datenverarbeiters erfolgt, hat ein Dateneigner zunächst kein Beweismittel an der Hand, durch das er einem B eindeutig nachweisen kann, dass eine Anfrage von ihm stammt. B könnte behaupten, A oder T hätten die Anfrage eingeschleust. Bei einem solchen Disput kann sich A jedoch an T wenden und durch ihn die vollständige Original-Anfrage inklusive der Signatur des B offenlegen lassen. Mit dieser lässt sich sodann beweisen, dass die Anfrage tatsächlich von B kam, da nur dieser im Stande ist, die Signatur zu erzeugen.

Die Datenverarbeiter B können anhand der veröffentlichten Liste und ihrer eigenen Aufzeichnungen zugleich überprüfen, ob T ihre Anfragen korrekt protokolliert hat.

6 Zusammenfassung und weiterer Forschungsbedarf

Mit dem vorgestellten Protokoll gelingt es dem Dateneigner, eine anonyme Datenablage mit „Sticky Policies“ vorzunehmen. Den Zeitpunkt wählt er selbst, um die konkrete Datenherausgabe an einen legitimierten Verarbeiter zum richtigen Anlass muss er sich nicht kümmern. Die Verwaltung der Gruppenmitgliedschaften obliegt einem Zertifizierer, der zusammen mit dem Treuhänder-Service auch den Schlüsselaustausch betreut. Das Protokoll gewährt den beiden Dienstleistern nur wenig Einblick in die Kommunikation – es werden die Abfragen der Verarbeiter wahrgenommen – über Dateneigner, den Inhalt der Policies oder der Datensätze erfahren sie nichts. Die Abfragen und Datenübermittlungen sind vollständig protokolliert und nicht abzustreiten.

Damit sind die Anforderungen an die Dienstleister relativ gering. Gegen die Neugierde von Treuhänder und Zertifizierer ist das System gewappnet, solange die beiden Parteien nicht böswillig kooperieren und die ihnen bekannten OTP miteinander teilen. Beide müssen jeweils ein grundsätzliches Interesse am Funktionieren des Systems haben, anderenfalls könnten sie es durch Verweigerung der Datenübermittlung sabotieren. Kompromittieren kann es jedoch keiner der beiden aus eigenen Stücken. Für das Gewinnen von Einsicht in die Datensätze sowie eine konstruktive Fälschung von Protokollen oder Inhalten wäre in jedem Fall die Kooperation mit mindestens einer weiteren der beteiligten Parteien notwendig.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

Der Schutz des Eigners vor Missbrauch seiner Daten steht im Vordergrund des Systems. Daneben ist ein zu untersuchendes Feld, wie auch die Interessen der Verarbeiter verstärkt durchgesetzt werden können. Insbesondere beruht der Nachweis aus einer Existenzabfrage auf der Beobachtung, ob zu einer bestimmten Policy ein Datensatz abgelegt wurde. Eine semantische Prüfung, ob der abgelegte Inhalt auch den Anforderungen der Policy genügt oder gar, ob er den wahren Verhältnissen des Eigners entspricht, findet nicht statt. Die Einführung einer zusätzlichen Instanz[18], die vor Verschlüsselung durch den Eigner die Daten einer Plausibilisierung unterzieht und nach Qualitätskriterien beglaubigt, ist ein möglicher Ansatz hierfür.

Der Dateneigner kann im Protokoll gegenüber Treuhänder und Zertifizierer anonym auftreten, während die Beobachtbarkeit und entsprechende Kontrollmöglichkeit der anderen Parteien im Sinne des Verfahrens ist. Benötigen die Verarbeiter im Anwendungsszenario einen starken Nachweis der Identität der Dateneigner, müsste die Signierfunktion S_A beispielsweise als fortgeschrittene oder qualifizierte elektronische Signatur[19] implementiert werden.

Die Einsatzmöglichkeiten des Protokolls sind über den Schutz rein personenbezogener Daten hinaus erweiterbar. Dazu sind lediglich die Definitionen der Datensatztypen und Verarbeitungszwecke dem gewählten Szenario anzupassen, sowie die Anforderungen an eine Zertifizierung als Mitglied der Verarbeitergruppen zu spezifizieren.

Bezieht man alle Arten von Daten ein, deren Aufdeckung gegenüber bestimmten Parteien für definierte Einsatzzwecke zu reservieren ist, findet sich etwa im Management von Vertragsbeziehungen ein weites Feld: Ein Lieferant verpflichtet sich vertraglich, für seinen Kunden ein Produkt mit spezifischen Eigenschaften herzustellen, und diese Eigenschaften beruhen auf der Anwendung von Verfahren oder Rezepturen, die als Unternehmensgeheimnisse nicht preisgegeben werden sollen. Dann kann man sich darauf einigen, die fraglichen Informationen gemäß dem geschilderten Protokoll bei einer Treuhänderinstanz zu hinterlegen, und die legitime Abfrage der Daten durch den Kunden auf die Zwecke zu beschränken, die sich aus einer möglichen Verletzung von Richtlinien zum Umweltschutz oder Patentansprüchen Dritter ergeben. So kann sich der Kunde gegenüber rechtlichen Risiken absichern, ohne dass der Lieferant seine Geschäftsgeheimnisse unnötig aufdeckt. Ebenso können Details von Preiskalkulationen, interne Bewertungen von Finanzprodukten oder Nachweise im Sinne des Geldwäschegesetzes die zu schützenden Daten darstellen.

Eine Mischform stellen Szenarien dar, die sowohl personenbezogene als auch von diesen abgeleitete Daten betreffen. Nimmt man exemplarisch die medizinische Forschung, bei der Forschungsinstitute Daten erheben und für Studienzwecke verarbeiten und aggregieren, treten die Institute zunächst gegenüber den Patienten als Verarbeiter und in einem zweiten Durchlauf des Verfahrens als Dateneigner auf. Sie räumen den Betroffenen, also den Teilnehmern der Studien, Zugriff zu detaillierten auf den Einzelnen bezogenen Erkenntnissen ein.

[18] Etwa in der Rolle eines „Auditors“, vgl. [SSB08].
[19] Im Sinne des Bundesgesetzes über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz).

ISBN 978-3-88579-286-4

So profitieren die Patienten von den Forschungsergebnissen. Gleichzeitig können die allgemeinen Ergebnisse der Studien, d.h. die statistischen Resultate, entgeltlich an Unternehmen weitergegeben werden, wenn diese sich entsprechend zertifiziert haben.

In der Weiterentwicklung des Systems ist zu ermitteln, wie weit das Risiko einer Kooperation von Treuhänder und Zertifizierer in betrügerischer Absicht reduziert werden kann, indem insbesondere verhindert wird, dass dem Treuhänder das beim Zertifizierer hinterlegte One-Time-Pad zugänglich gemacht wird.

Literaturverzeichnis

[Al07] Al-Fedaghi, Sabha: Dismantling the Twelve Privacy Purposes. In Trust management: Proceedings of IFIPTM 2007, Joint iTrust and PST Conferences on Privacy, Trust Management and Security, July 30-August 2, 2007, New Brunswick, Canada; S. 207–222, Springer Verlag, Boston, 2007.

[APS02] Ashley, Paul; Powers, Calvin; Schunter, Matthias: From Privacy Promises to Privacy Management. A New Approach for Enforcing Privacy Throughout an Enterprise. In (Hempelmann, Christian F.; Raskin, Victor, Hrsg.): Proceedings - New Security Paradigms Workshop 2002, September 23-26, Virginia Beach, VA, USA. ACM Press, New York, 2002, S. 43-50.

[De84] DeLaurentis, John M.: A further weakness in the common modulus protocol for the RSA cryptoalgorithm. In: Cryptologia, Volume VIII Number 3, Taylor & Francis, London, 1984, S. 253-259.

[FP02] Federrath, Hannes; Pfitzmann, Andreas: Technische Grundlagen. In (Rossnagel, Alexander, Hrsg.): Handbuch des Datenschutzrechts, Beck Verlag, München, 2002.

[Gr01] Greß, Sebastian: Datenschutzprojekt P3P, Darstellung und Kritik. In Datenschutz und Datensicherheit 25 (Mär. 2001) 3, Vieweg, Wiesbaden, 2011, S. 144-149.

[Ku07] Kumaraguru, Ponnurangam; Cranor, Lorrie Faith; Lobo, Jorge; Calo, Seraphin B.: A Survey of Privacy Policy Languages. Workshop on Usable IT Security Management (USM '07), http://cups.cs.cmu.edu/soups/2007/workshop/Privacy_Policy_Languages.pdf

[Mö06] Möller, Jan: Automatisiertes Management von Datenschutzrechten. In Datenschutz und Datensicherheit 30 (Feb. 2006) 2, Vieweg, Wiesbaden, 2006, S. 98-101.

[Ri10a] Rijmenants, Dirk: Is One-time Pad History? In Cipher Machines and Cryptology, http://users.telenet.be/d.rijmenants/papers/is_one_time_pad_history.pdf.

[Ri10b] Rijmenants, Dirk: The complete guide to secure communications with the one time pad cipher In Cipher Machines and Cryptology, http://users.telenet.be/d.rijmenants/papers/one_time_pad.pdf.

[Sc96] Schneier, Bruce: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C. 5. Auflage, Addison-Wesley, Bonn, 1996.

[Sc09] Schmeh, Klaus: Kryptografie: Verfahren, Protokolle, Infrastrukturen. 4., aktualisierte und erweiterte Auflage, dpunkt-Verlag, Heidelberg, 2009.

[Si83] Simmons, Gustavus James: A "Weak" Privacy Protocol Using the RSA Crypto Algorithm. In: Cryptologia, Volume VII Number 2, Taylor & Francis, London, 1983, S. 180-182.

[SSB08] Shah, Mehul A.; Swaminathan, Ram; Baker, Mary: Privacy-Preserving Audit and Extraction of Digital Contents. HP Laboratories, Palo Alto, http://www.hpl.hp.com/techreports/2008/HPL-2008-32R1.pdf.

[TW06] Talbot, John; Welsh, Dominic: Complexity and Cryptography. Cambridge University Press, New York, 2006.

ISBN 978-3-88579-286-4