Mitt. Math. Ges. Hamburg **34** (2014), 21–32

Informationssicherheit und technischer Datenschutz durch verteilte Systeme

Hannes Federrath

Zusammenfassung

Als Schutzziele für die Sicherheit von informationstechnischen Systemen gelten seit mindestens 30 Jahren Vertraulichkeit, Integrität und Verfügbarkeit. Moderne Verfahren der Informationssicherheit sind – ebenso wie die Kommunikationssysteme selbst – heute meist als verteilte Systeme ausgestaltet. Verteiltheit bedeutet einerseits Mehrfachauslegung (Redundanz) von Systemteilen zur Verbesserung der Verfügbarkeit, aber auch Diversität (Verschiedenartigkeit der Herkünfte) zur Verbesserung der Sicherheit vor systematischen Fehlern aber auch zum Schutz vor trojanischen Pferden (Schutz vor verdeckten Kanälen). Außerdem bieten kryptographische Bausteine wie das DC-Netz und das Mix-Netz Möglichkeiten zum Schutz der Anonymität von Netzteilnehmern, die ohne Verteiltheit gar nicht realisierbar wären.

1 Einführung

Heute gelten die drei Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit als eingängige, weil leicht verständliche und dennoch vollständige Anforderungen an sichere IT-Systeme. Sie lassen sich ableiten aus den drei Bedrohungen

- unbefugter Informationsgewinn,
- unbefugte Modifikation von Daten und
- unbefugte Beeinträchtigung der Funktionalität,

die Vodydock und Kent [1] formuliert haben. Vertraulichkeit, Integrität und Verfügbarkeit adressieren im Wesentlichen Risiken, die durch regelwidriges Verhalten in IT-Systemen entstehen. Daher werden sie gelegentlich auch als *klassische Schutzziele der IT-Sicherheit* bezeichnet. Allerdings ist diese Sicht etwas einseitig.

IT-Systeme sollten zunächst so gestaltet werden, dass regelwidriges Verhalten dank der eingesetzten Sicherheitsmechanismen wenigstens erkennbar ist, besser noch verhindert wird. Selbst wenn zwischen den Beteiligten ein Konsens darüber existiert, welche Handlungen erlaubt sein sollen und welche nicht, kann es dennoch unerfüllbare Schutzanforderungen (wenigstens aus Sicht eines Beteiligten) geben. Wünscht beispielsweise ein Beteiligter die zweifelsfreie Identifizierung seiner Kommunikationspartner (Schutzziel Integrität), können diese nicht gleichzeitig anonym bleiben (Schutzziel Vertraulichkeit).

Die Erkennung und Aushandlung gegensätzlicher Schutzanforderungen und die Schaffung von Lösungen, die die Sicherheitsinteressen aller Beteiligten berücksichtigen, ist Aufgabe der **mehrseitigen Sicherheit**, die in wesentlichen Teilen von Müller und Pfitzmann [2] postuliert wurde. Pfitzmann und Wolf [3] haben die Abhängigkeiten zwischen den Schutzzielen der mehrseitigen Sicherheit untersucht und herausgearbeitet, an welchen Stellen überhaupt zwischen den Beteiligten gegensätzliche Sicherheitsinteressen auftreten können. Rost und Pfitzmann [4] haben u.a. aufbauend auf den Arbeiten von [3] die Liste der Schutzziele um einige Begriffe (Unentdeckbarkeit, Kontingenz, Abstreitbarkeit, Verbindlichkeit, Findbarkeit, Ermittelbarkeit) erweitert. Die derzeit umfassendste Auflistung von Schutzzielen mit Bezug zur mehrseitigen Sicherheit und zum **Datenschutz** haben Rost und Bock [5] vorgestellt (siehe auch Abbildung 1).

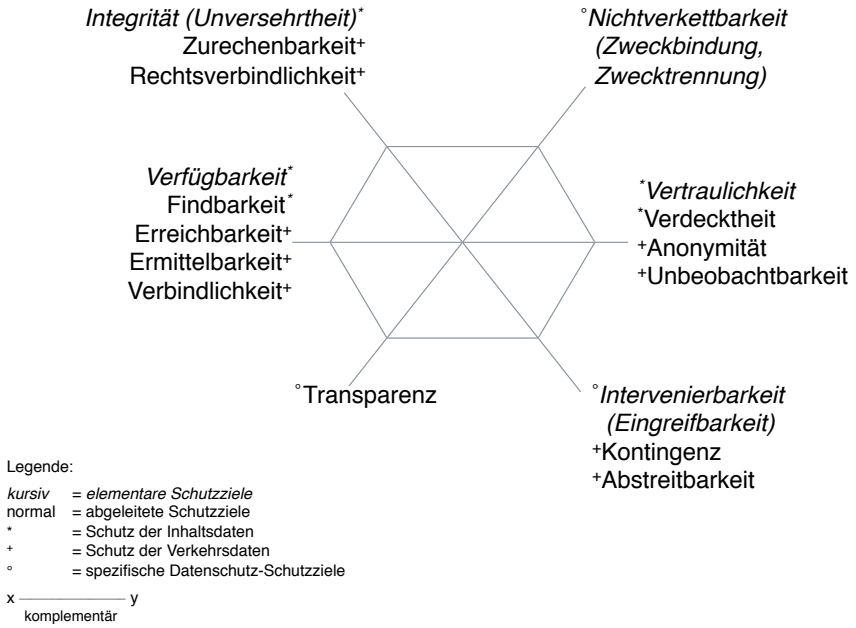

Abbildung 1: Schutzziele in Anlehnung an [5]

Im Folgenden sollen nun entlang der drei Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit einige ausgewählte Möglichkeiten zum Schutz durch Verteiltheit und verteilte Systeme vorgestellt werden.

2 Verfügbarkeit

Im Bereich der Verfügbarkeit existieren zwei grundsätzliche Techniken zum Schutz. Erstens kann durch **Redundanz**, d.h. die mehrfache Auslegung von Sy-

stemkomponenten dafür gesorgt werden, dass bei Ausfall einer Komponente das Gesamtsystem weiterhin verfügbar ist: Die redundante Komponente übernimmt die Funktion der ausgefallenen Komponente. Redundanz ist insbesondere bei erwartbaren Ausfällen durch Verschleiß und Alterung eine geeignete und sehr weit verbreitete Schutzmaßnahme. Sie hilft jedoch nicht bei Konstruktionsfehlern, insbesondere dann, wenn diese erst spät und während des Betriebs erkannt werden.

Deshalb müssen sichere und zuverlässige IT-Systeme zweitens durch **Diversität**, d.h. Verschiedenartigkeit der Herkünfte, gegen unerkannte systematische Konstruktionsfehler geschützt werden. Wenn beispielsweise eine redundant, aber nicht diversitär ausgelegte Serverkomponente, z.B. ein Webserver, aufgrund eines Programmierfehlers (Softwarefehler) ausfällt, wird auch die Ersatzkomponente ausfallen: Sie enthält den selben Programmierfehler und stürzt ebenfalls ab. Wären auf den redundant ausgelegten Hardware-Servern verschiedene Softwaresysteme (z.B. Webserver unterschiedlicher, unabhängiger Hersteller) installiert, wäre die Ausfallwahrscheinlichkeit aufgrund eines Software-Konstruktionsfehler vermutlich geringer.

Diversität ist (in gewissem Sinn) das Gegenteil von Monokultur und hat sogar einen positiven Einfluss auf das Schutzziel Vertraulichkeit. Wenn ein großes Gesamtsystem durchgehend von einem einzigen Hersteller stammt, steigt die Wahrscheinlichkeit, dass entweder viele Systemkomponenten dieses Herstellers gleichartige Fehler enthalten, die von Angreifern ausgenutzt werden könnten, um Information über verdeckte Kanäle und/oder trojanische Pferde zu gewinnen (vgl. [6]). Diversitär gestaltete Systeme machen es dem Angreifer hier ggf. deutlich schwerer, das Schutzziel Vertraulichkeit zu verletzen.

3 Integrität

Auch im Bereich Integrität (Schutz vor unbefugter Modifikation) kann Verteiltheit die Sicherheit erhöhen. Eine triviale Möglichkeit zum Schutz vor Übertragungsfehlern wäre beispielsweise die mehrfache Übertragung von Daten auf unterschiedlichen Übertragungswegen. Allerdings gelingt durch den Einsatz kryptographischer Verfahren zumindest die Erkennung von (absichtlichen) Datenveränderungen weitaus zuverlässiger und effizienter. Verteiltheit zum Schutz vor unabsichtlichen Übertragungsfehlern oder absichtlichen Übertragungsstörungen ist daher eher unüblich.

Bei den sog. Public Key Infrastructures (PKI) wird das Prinzip der Verteiltheit im Bereich der Integrität jedoch als Architekturprinzip breit angewendet. PKIs kommen werden zur Echtheitsprüfung von öffentlichen Schlüsseln in asymmetrischen Kryptosystemen eingesetzt. Praktische Anwendung finden PKIs beispielsweise bei Secure Sockets Layer (SSL, erkennbar am https beim Websurfen) und beim E-Mail-Verschlüsselungsstandard S/MIME (Secure Multipurpose Internet Mail Extensions).

Eine PKI ist ein hierarchischer Graph (Baum) von gerichteten Vertrauensbeziehungen zwischen sog. Certification Authorities (CA). Auf der untersten Ebene des Baumes befinden sich als Blattknoten die Endnutzer, die von einer direkt übergeordneten Zertifizierungsstelle (CA) eine Beglaubigung (Zertifikat) ihres öffentlichen Schlüssels erhalten. Jede CA verfügt selbst über einen öffentlichen Schlüssel, mit dem die von der CA beglaubigten öffentlichen Nutzerschlüssel prüfbar sind. Der öffentliche Schlüssel einer CA wird von der hierarchisch übergeordneten CA auf die gleiche Weise wie die öffentlichen Nutzerschlüssel beglaubigt.

Möchte ein Nutzer die Echtheit eines (fremden) öffentlichen Nutzerschlüssels feststellen, muss er alle Beglaubigungen von unten nach oben im Baum prüfen, solange, bis er auf ein CA-Zertifikat trifft, dessen Echtheit er bereits festgestellt hat.

Da alle Beglaubigungen im Baum spätestens bei einem gemeinsamen Wurzelknoten zusammenlaufen, existiert für alle Nutzer, die selbst innerhalb der PKI einen beglaubigten Schlüssel besitzen, stets ein Pfad zum Prüfen der Echtheit von fremden Nutzerschlüsseln, dessen gemeinsamer „Vertrauensanker“ spätestens der öffentliche Schlüssel der Wurzel-CA ist.

Was in der Theorie sicher erscheint, hat sich insbesondere nach den NSA-Enthüllungen [7, 8] durch Edward Snowden als in der Praxis ungenügend herausgestellt. Es muss allen Zertifizierungsstellen entlang eines Zertifizierungspfades vertraut werden. Wenn die Beglaubigung eines öffentlichen CA-Schlüssels oder öffentlichen Nutzerschlüssels (unabsichtlich) falsch oder (absichtlich) gefälscht ist, müssen alle Beglaubigungen, die durch diese CA erteilt wurden ungültig gemacht werden.

Daher liegt es nahe, den Grad der Verteiltheit innerhalb einer PKI durch Hinzunahme weiterer Kanten zu erhöhen. Es existieren dann gewissermaßen mehrere „Vertrauenspfade“ zur Überprüfung der Echtheit der öffentlichen Schlüssel, wobei bereits genau ein einziger tatsächlich vertrauenswürdiger Pfad genügt, um die Echtheit festzustellen. Dies ist beispielsweise beim **Web-of-Trust** der Fall, wo sich die Nutzer gegenseitig ihre öffentlichen Schlüssel beglaubigen und auf ein hierarchisches Zertifizierungsmodell verzichtet wird. Das Web-of-Trust wird beispielsweise bei OpenPGP angewendet. Auch innerhalb eines hierarchischen Zertifizierungsmodells wird in der Praxis die Anzahl der Kanten erhöht, allerdings nur zwischen den CAs. Dann spricht man von **Cross Certification**. Neuere Ansätze wie **Laribus** [9] erlauben auch die erweiterte Echtheitsprüfung im hierarchischen Zertifizierungsmodell anhand von Kontextinformationen, die zwischen Nutzern ausgetauscht werden.

4 Vertraulichkeit

Vertraulichkeit bedeutet Schutz vor unbefugtem Informationsgewinn. Es liegt auf der Hand, dass die Datenverschlüsselung ein geeigneter Mechanismus zum Schutz der Vertraulichkeit von **Inhaltsdaten** ist. Dies gilt sowohl bei der Datenspeicherung (z.B. Festplattenverschlüsselung) als auch bei der Datenkommunikation (z.B. E-Mail-Verschlüsselung, allgemein bestenfalls als sog. Ende-zu-Ende-Verschlüsselung). Eine weiter gehende Form der Vertraulichkeit von Inhaltsdaten wird durch Steganographie (Schutzziel Verdecktheit) erreicht. Hier wird neben dem Inhalt auch die Existenz eines Inhalts verborgen (siehe auch Abbildung 2).

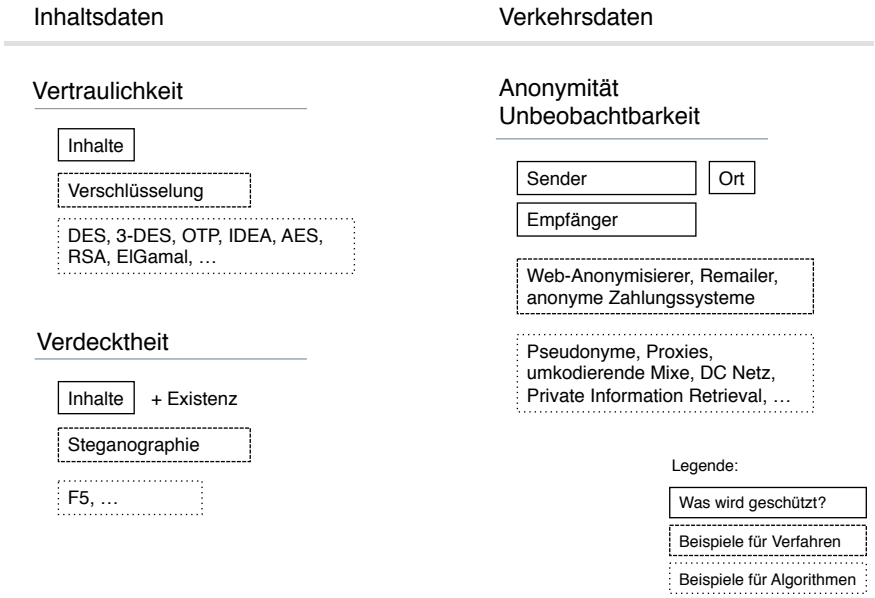

Abbildung 2: Schutzziel Vertraulichkeit: Verfahren und Algorithmen

Bei jeder Kommunikation fallen allerdings auch **Verkehrsdaten** an: Sie ermöglichen den Informationsgewinn beim Angreifer darüber, wer mit wem wie lange und ggf. an welchem Ort kommuniziert. Um die Schutzziele der Vertraulichkeit hinsichtlich der Verkehrsdaten präzise formulieren zu können, haben sich die Begriffe Anonymität und Unbeobachtbarkeit etabliert. Anonymität betrifft den Schutz der Identität einer kommunizierenden Instanz (Sender und/oder Empfänger) mindestens gegenüber dem jeweiligen Kommunikationspartner. Unbeobachtbarkeit betrifft die Unverkettbarkeit von Kommunikationsereignissen (Senden und/oder Empfangen) vor allen Außenstehenden und/oder allen Netzbetreibern.

Datenverschlüsselung (Schutzziel Vertraulichkeit der Inhaltsdaten) bietet leider nur eingeschränkten Schutz der Anonymität und Unbeobachtbarkeit. Wenn alle Leitungen zwischen allen Netzbetreibern durch Datenverschlüsselung geschützt werden (sog. Verbindungsverschlüsselung), können außenstehende Angreifer tatsächlich keine Information mehr gewinnen; allerdings müssen bei den Netzbetreibern zumindest die Adressdaten im Klartext vorliegen, da ansonsten kein Routing möglich wäre. Die Netzbetreiber erfahren also Verkehrsdaten, weil sie sie meist zur Diensterbringung benötigen.

Verteiltheit ist insbesondere ein geeignetes Konzept zum Schutz der Anonymität und Unbeobachtbarkeit gegenüber Netzbetreibern. Die wichtigsten Grundverfahren – DC-Netze [10], der Blind-Message-Service [11], sowie Mix-Netze [12] – machen ausnahmslos von Verteiltheit Gebrauch. Die wesentliche Idee dabei ist, dass durch die Verteiltheit mehr als eine Instanz angreifen muss, um vertrauliche Information (hier: Verkehrsdaten) zu gewinnen.

DC-Netz

Beim DC-Netz [10] werden Broadcast, Verschlüsselung und Dummy Traffic miteinander kombiniert, um die **Anonymität des Senders** zu schützen. Senderanonymität kann nur erreicht werden, wenn mehrere Sender gleichzeitig aktiv sind. Das DC-Netz arbeitet rundenbasiert. In jeder Runde wird von jedem Sender eine Nachricht mit festgelegter Länge erwartet. Alle Nachrichten aller Sender werden an alle Empfänger übermittelt (sog. Broadcast), d.h. es findet keine Punkt-zu-Punkt-Kommunikation statt. Üblicherweise sind die Sender auch gleichzeitig die Empfänger. Da alle Sender jeweils pro Runde genau eine Nachricht beisteuern, kann ein Angreifer keinerlei Information darüber gewinnen, ob ein Sender tatsächlich etwas senden möchte. Sender ohne echte Nachricht senden Leernachrichten (Dummy Traffic). Eine Leernachricht ist im DC-Netz stets eine Folge von Null-Bits. Um Leernachrichten von echten Nachrichten für den Angreifer ununterscheidbar zu machen, sind alle Nachrichten (sowohl echte als auch leere) verschlüsselt. Hier kommt eine Überlagerung (bitweise XOR-Verknüpfung) mit echt zufälligen Bitfolgen zum Einsatz.

Wollen beispielsweise drei Sender A, B und C während des Sendens anonym bleiben, schließen sie sich in einem DC-Netz zusammen (siehe Abbildung 3). A, B und C tauschen vorab paarweise Schlüssel miteinander aus. Jeder Schlüssel ist eine echt zufällige Bitfolge, die nur ein einziges Mal verwendet werden darf – deswegen manchmal auch One-Time-Pad genannt, aber nicht zu verwechseln mit dem gleichnamigen informationstheoretisch sicheren Verschlüsselungsverfahren – und der festgelegten Nachrichtenlänge pro Runde entspricht. Nach dem Schlüsselaustausch besitzt jeder Sender pro Runde jeweils zwei Schlüssel: A besitzt (k_{AB}, k_{AC}), B besitzt (k_{AB}, k_{BC}), C besitzt (k_{AC}, k_{BC}).

Angenommen Sender A möchte eine anonyme Nachricht senden, während B und C nur Leernachrichten senden wollen. Dann überlagert (XOR-verknüpft)

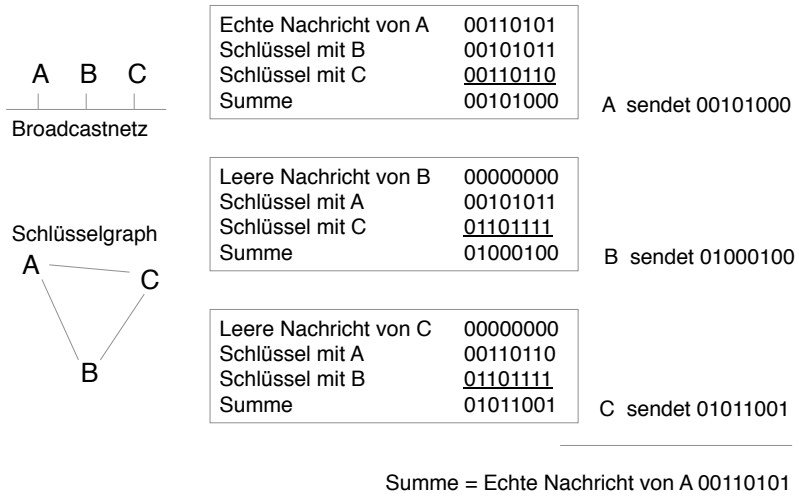

Abbildung 3: Überlagerndes Senden zum Schutz der Senderanonymität

jeder der drei Sender geheim seine Nachricht (echt oder leer) mit den beiden Schlüsseln und sendet sein Ergebnis (sog. lokale Summe) an die beiden anderen Sender. Jeder Sender überlagert (XOR-verknüpft) die drei lokalen Summen und erhält als Ergebnis die Nachricht des (anonymen) Senders.

Das DC-Netz gewährleistet unbedingte Anonymität (Schutzziel Vertraulichkeit des Senders), wenn die beiden Sender der Leernachrichten nicht gemeinsam angreifen. Genauer gesagt bleibt der Sender innerhalb der Gruppe der Sender anonym, die über wenigstens jeweils einen für den Angreifer unbekannten Schlüssel miteinander verbunden sind (sog. Schlüsselgraph). Das verteilte System wird gewissermaßen durch die Teilnehmer selbst gebildet.

Offensichtlich funktioniert das DC-Netz nur, wenn alle Sender kooperativ handeln, d.h. sich an das Protokoll halten (Schutzziel Verfügbarkeit). Zudem dürfen sich nicht mehrere Sendewünsche pro Runde miteinander überlagern. Zwar kann man derartige Kollisionen unter Wahrung der Anonymität aller Sender auflösen [6], in der Praxis findet man das DC-Netz allerdings bisher kaum, da die Verfügbarkeitsprobleme des DC-Netzes kaum beherrschbar sind. Hier zeigt sich beispielsweise sehr schön, dass einige Schutzziele (hier: Verfügbarkeit und Anonymität) komplementär zueinander sind (siehe auch Abbildung 1).

Blind-Message-Service

Der Blind-Message-Service [11] ermöglicht unbeobachtbare Datenbankabfragen aus von unabhängigen Betreibern replizierten Datenbanken und dient somit der **Anonymität des Empfängers** (Clients). Auch beim Blind-Message-Service

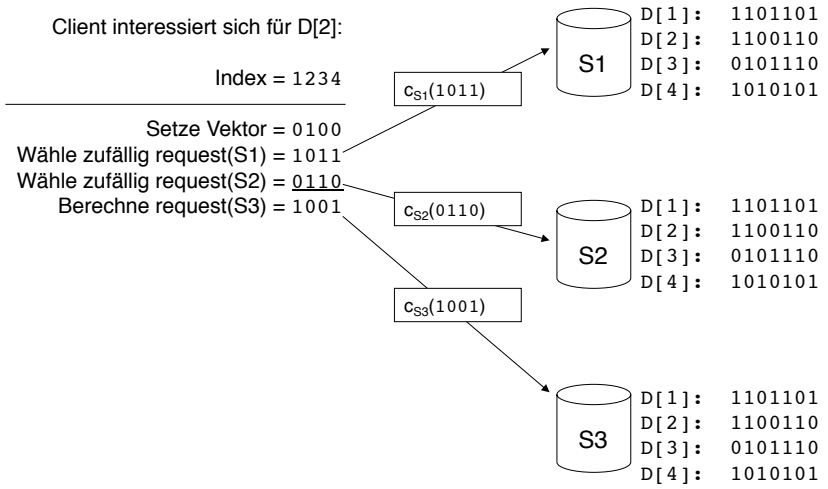

Abbildung 4: Blind-Message-Service: Anfrage

kommt eine Überlagerung (bitweise XOR-Verknüpfung) mit echt zufälligen Bitfolgen zum Einsatz: Eine Datenbankanfrage (genauer: Zugriff auf bestimmten Datensatz D_i) wird als linearer Bitvektor modelliert. Das verteilte System zum Schutz der Anonymität wird hier aus n mit $n > 1$ replizierten Datenbankservern gebildet. Um den Datensatz D_i abzufragen, werden n für den Angreifer zufällig aussehende Abfragevektoren gebildet. Die n Abfragevektoren werden vom Client so gebildet, dass deren bitweise XOR-Verknüpfung genau einen linearer Bitvektor bildet, für den der Index i ein Eins-Bit ist und ansonsten nur Null-Bits enthält (siehe Abbildung 4).

Jede Datenbank erhält genau einen der n Abfragevektoren. Die Datenbanken erfahren nicht, für welchen Datensatz sich der Client interessiert, solange wenigstens einer der n Datenbankbetreiber seinen Abfragevektor geheim hält, d.h. maximal $n-1$ Datenbankbetreiber dürfen gemeinschaftlich angreifen, was ihnen jedoch keinen Informationsgewinn ermöglicht.

Jede Datenbank überlagert nun genau die Datensätze, für die im Abfragevektor ein Eins-Bit gesetzt ist (lokale Summe). Deshalb müssen alle Datensätze auf die gleiche Länge normiert sein. Längere Datensätze können in mehrere Datensätze zerlegt werden, die dann nacheinander abgefragt werden müssen.

Alle n lokalen Summen (siehe Abbildung 5 mit $n = 3$) werden an den Client übermittelt, der sie seinerseits überlagert und als Ergebnis den gewünschten Datensatz erhält.

Natürlich muss alle Kommunikation zwischen Client und den Datenbanken Ende-zu-Ende-verschlüsselt sein, damit Außenstehende nicht einfach alle Kommunikation mitlesen und das Überlagungerungsergebnis berechnen können. Da

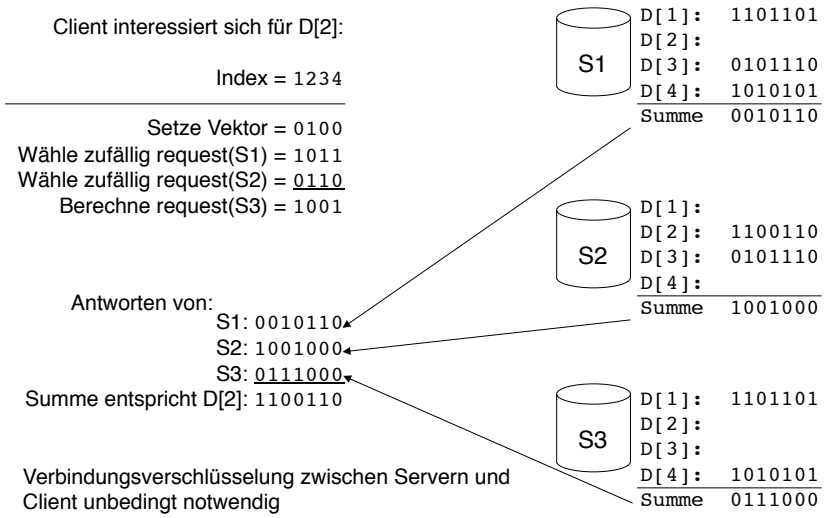

Abbildung 5: Blind-Message-Service: Antwort

auch die Datenbanken nicht alle lokalen Summen kennen (solange maximal $n-1$ Datenbankbetreiber gemeinschaftlich angreifen), können auch die Datenbanken keine Information über den angefragten Datensatz gewinnen.

Hinsichtlich der Anonymität (Schutz vor Beobachtung durch die Datenbanken) ist der Sicherheitsparameter n (Anzahl der Datenbanken und damit Grad der Verteiltheit) entscheidend. Je größer n gewählt wird, umso unwahrscheinlicher wird es, dass gemeinschaftliche Angriffe aller Datenbankbetreiber erfolgreich sind.

Mix-Netz

Das Mix-Netz [12] dient dem **Schutz der Kommunikationsbeziehung** zwischen einem Sender und Empfänger. Hierzu werden n mit $n > 1$ spezielle Router (sog. Mixe) zwischengeschaltet. Mixe arbeiten wie Proxies, d.h. sie leiten eingehende Nachrichten weiter und verbergen somit die Adresse des Senders (siehe Abbildung 6 mit drei Sendern S, drei Empfängern E und $n = 3$ Mixen in einer sog. Mix-Kaskade).

Mixe arbeiten unter einem deutlich stärkeren Angreifermodell als Proxies: Die Angreifer im Mix-Netz dürfen alle Kommunikationsleitungen überwachen. Außerdem dürfen maximal $n-1$ Mixe gemeinschaftlich angreifen. Solange wenigstens ein einziger Mix vertrauenswürdig ist, bleibt auch die Kommunikationsbeziehung zwischen Sender und Empfänger unbeobachtbar. Die NSA-Enthüllungen zeigen, dass die globale Überwachung und langfristige Speicherung von Verkehrsdaten

im Internet stattfindet. Das starke Angreifermodell des Mix-Netzes ist also keineswegs übertrieben.

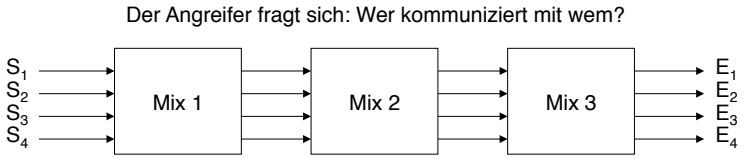

Abbildung 6: Beispiel für eine Mix-Anordnung

Mixe sorgen dafür, dass die ein- und ausgehenden Mix-Nachrichten unverkettbar sind. Hierzu sammelt ein Mix eingehende Nachrichten vieler Absender in einem sog. Schub, ändert das „Aussehen“ jeder Nachricht und gibt die gesammelten Nachrichten in geänderter Reihenfolge aus. Das „Aussehen“ einer Nachricht wird geändert, indem eingehende Nachrichten mit dem öffentlichen Schlüssel des Mixes verschlüsselt sind und vom Mix entschlüsselt werden. Auch beim Mix-Netz ist die Nachrichtenlänge pro Schub für alle Nachrichten gleich. Da Mixe deterministisch arbeiten, würde eine vom Angreifer wiederholt eingespielte Nachricht eine gleiche ausgehende Nachricht erzeugen und somit verkettbar machen. Daher darf ein Mix jede eingehende Nachricht nur genau einmal ausgeben (sog. Replay-Erkennung über alle Schübe hinweg).

Die praktisch verfügbaren Internet-Anonymisierer TOR [13] und AN.ON [14] arbeiten grundsätzlich nach dem Mix-Prinzip. Zugunsten der Performance wird jedoch das Sammeln von Nachrichten auf ein gerade noch vertretbares Maß reduziert. Das Sammeln von Nachrichten macht insbesondere das Websurfen über die Internet-Anonymisierer langsam.

Was die Rechtmäßigkeit des Betriebs von Internet-Anonymisierern betrifft, enthält das Telemediengesetz (TMG) sogar eine Verpflichtung zur Ermöglichung von Anonymität: „Der Diensteanbieter hat dem Nutzer die Inanspruchnahme von Telediensten und ihre Bezahlung anonym oder unter Pseudonym zu ermöglichen, soweit dies technisch möglich und zumutbar ist. Der Nutzer ist über diese Möglichkeit zu informieren.“ (§ 13 Abs. 6 TMG). Allerdings waren in der kurzen Zeitspanne von 2008 bis 2010 u.a. deutsche Anbieter von Internet-Anonymisierern durch das „Gesetz zur Neuregelung der Telekommunikationsüberwachung und anderer verdeckter Ermittlungsmaßnahmen sowie zur Umsetzung der Richtlinie 2006/24/EG“ vom 21. Dezember 2007 (BGBl. I 2007, S. 3198 ff.) dazu verpflichtet, 6 Monate auf Vorrat alle Verbindungsdaten zu speichern. Im Jahr 2010 wurde die deutsche Umsetzung der Richtlinie 2006/24/EG über die Vorratsdatenspeicherung vom Bundesverfassungsgericht für verfassungswidrig erklärt. Derzeit beschäftigt sich der Europäische Gerichtshof (EuGH) mit der Frage, ob die Richtlinie gegen europäisches Recht verstößt.

5 Schlussbemerkungen

Die Komponenten verteilter Systeme zum Schutz der Vertraulichkeit werden von unterschiedlichen Betreibern bereitgestellt. Verteilte Systeme wie zum Beispiel die Mixe und der Blind-Message-Service arbeiten einerseits kooperativ an der Diensterbringung (hier: dem Schutz der Anonymität) und realisieren gleichzeitig das Prinzip der Gewaltenteilung, indem jeder Betreiber nur einen Teil der Verbindung kennt. Einem globalen Überwacher wird es dadurch extrem schwer gemacht, die Kommunikationsbeziehungen aller Internet-Nutzer zu beobachten. Verteilte Systeme schützen somit auch Grundrechte wie die informationelle Selbstbestimmung und die Meinungsfreiheit.

Verteiltheit kann auch vor versteckten Ausspähfunktionen in Soft- und Hardware schützen: Wenn interoperable Systeme verschiedener Hersteller über wohldefinierte und offen gelegte Schnittstellen kommunizieren, wird es einem Angreifer schwer fallen, flächendeckend „Implantate“ in Soft- und Hardware zur Überwachung zu nutzen, da verdeckte Kanäle erheblich erschwert werden. Das Bundesverfassungsgericht hat bereits im Jahr 2008 in einer Entscheidung zur „Online-Durchsuchung“ ein Grundrecht auf Gewährleistung der Vertraulichkeit und Integrität informationstechnischer Systeme aus dem allgemeinen Persönlichkeitsrecht abgeleitet [15]. Mit den NSA-Enthüllungen ist dieses Grundrecht aktueller denn je.

Verteilte Systeme sind Basisbausteine heutiger Kommunikationssysteme. Zusammenfassend kann gezeigt werden, dass verteilte Systeme nicht nur einen Beitrag zur schnellen und zuverlässigen Kommunikation leisten, sondern auch die Vertraulichkeit von Daten schützen können.

Literatur

[1] Viktor L. Voydock, Stephen T. Kent: Security Mechanisms in High-Level Network Protocols. ACM Computing Surveys 15/2 (1983) 135–170.

[2] Günter Müller, Andreas Pfitzmann (Hg.): Mehrseitige Sicherheit in der Kommunikationstechnik. Addison-Wesley-Longman, 1997.

[3] Gritta Wolf, Andreas Pfitzmann: Properties of protection goals and their integration into a user interface. Computer Networks 32/6 (2000) 685–700.

[4] Martin Rost, Andreas Pfitzmann, Datenschutz-Schutzziele – revisited. Datenschutz und Datensicherheit DuD 33/6 (2009) 353–358.

[5] Martin Rost, Kirsten Bock: Privacy By Design und die Neuen Schutzziele – Grundsätze, Ziele und Anforderungen. Datenschutz und Datensicherheit DuD 35/1 (2011) 30-35.

[6] Andreas Pfitzmann: Sicherheit in Rechnernetzen: Mehrseitige Sicherheit in verteilten und durch verteilte Systeme. Vorlesungsskript. TU Dresden, 1999. http://dud.inf.tu-dresden.de/~pfitza/DSuKrypt.pdf

[7] Bruce Schneier: New NSA Leak Shows MITM Attacks Against Major Internet Services. Sep 13, 2013. https://www.schneier.com/blog/archives/2013/09/new_nsa_leak_sh.html

[8] InformationWeek: Stolen Digital Certificates Compromised CIA, MI6, Tor. Sep 6, 2011. http://www.informationweek.com/attacks/stolen-digital-certificates-compromised-cia-mi6-tor/d/d-id/1099964

[9] Andrea Micheloni, Karl-Peter Fuchs, Dominik Herrmann, Hannes Federrath: Laribus: Privacy-Preserving Detection of Fake SSL Certificates with a Social P2P Notary Network. 8th International Conference on Availability, Reliability and Security (ARES), Regensburg, Sep 2-6, 2013.

[10] David Chaum: The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. Journal of Cryptology 1 (1988) 65-75.

[11] David A. Cooper, Kenneth P. Birman: Preserving privacy in a network of mobile computers. In: 1995 IEEE Symposium on Research in Security and Privacy. IEEE Computer Society Press, Los Alamitos, 1995, 26-38. http://cs-tr.cs.cornell.edu:80/Dienst/UI/1.0/Display/ncstrl.cornell/TR85-1490

[12] David Chaum: Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. Communications of the ACM 24/2 (1981) 84-88.

[13] Tor Project: Anonymity Online. http://www.torproject.org

[14] Projekt: AN.ON - Anonymität.Online. http://www.anon-online.de

[15] BVerfG, 1 BvR 370/07 vom 27.2.2008, Absatz-Nr. (1 - 333), http://www.bverfg.de/entscheidungen/rs20080227_1bvr037007.html

Eingegangen am 29. 03. 2014

Hannes Federrath
Fachbereich Informatik
Universität Hamburg
https://svs.informatik.uni-hamburg.de