„Neue“ Anonymitätstechniken

Eine vergleichende Übersicht

Hannes Federrath, Andreas Pfitzmann

Der Beitrag bewertet die Sicherheit von Crowds, Anonymizer und Onion Routing, also Systemen zur Unterstützung von Client-Anonymität im World Wide Web. Je nach unterstellter Stärke des Angreifers kann sich ein Nutzer mit diesen Techniken gezielt schützen.

[FOTO] Dr. Hannes Federrath

Wissenschaftlicher Mitarbeiter an der TU Dresden, Fakultät Informatik, Forschungsschwerpunkt ist die Sicherheit in verteilten Systemen

E-Mail: federrath@inf.tu-dresden.de

[FOTO] Prof. Dr. Andreas Pfitzmann

TU Dresden, Fakultät Informatik, Forschungsschwerpunkt ist technischer Datenschutz durch verteilte Systeme

E-Mail: pfitza@inf.tu-dresden.de

Einleitung

Zunächst werden einige zum Verständnis der vorgestellten Anonymitätstechniken wichtige Begriffe wie Anonymität, Unbeobachtbarkeit und Unverkettbarkeit definiert. Anschließend werden die bekannten Standardtechniken systematisiert und kurz vorgestellt. Am Beispiel der Mixe wird gezeigt, wie solche Techniken funktionieren.

In Abschnitt 2 wird das Angreifermodell beschrieben, das der Bewertung der anschließend vorgestellten Verfahren zugrunde liegt. In Abschnitt 3 werden die Verfahren vorgestellt und bezüglich ihrer erreichten Sicherheit verglichen.

1.1 Begriffe

Anonymität und Unbeobachtbarkeit sind Forderungen nach Vertraulichkeit von Daten. **Vertraulichkeit** allgemein kann sich auf zwei Bereiche beziehen. Es können sowohl

- die Nachrichteninhalte als auch
- die sogenannten Kommunikationsumstände

schützenswert sein.

Die Vertraulichkeit von Nachrichteninhalten erreicht man normalerweise durch Verschlüsselung. Schützenswerte **Kommunikationsumstände** können sein:

- das *Senden* einer Nachricht,
- das *Empfangen* einer Nachricht,
- eine *Kommunikationsbeziehung* zwischen zwei oder mehr Teilnehmern,
- der *Aufenthaltsort* der Teilnehmer.

Dabei ist in der Regel jeweils noch der Zeitpunkt des Vorgangs ein wichtiger Umstand. Bei jeder Kommunikation fallen nun sogenannte Verkehrsdaten (Beginn und Ende der Kommunikation, eindeutige Kennungen der Kommunikationspartner, Umfang der Kommunikation) an, die einem Angreifer Informationen über diese Kommunikationsumstände geben können.

In [Pfit_90] wurden Definitionen für die Begriffe Unbeobachtbarkeit und Anonymität angegeben, die hier in etwas vereinfachter Form verwendet werden sollen.

Unbeobachtbarkeit bedeutet, daß für einen Angreifer die Wahrscheinlichkeit des Auftretens eines Ereignisses (z. B. Senden einer Nachricht) *nach* jeder Beobachtung sowohl echt größer 0 als auch echt kleiner 1 ist. Perfekte Unbeobachtbarkeit bedeutet, daß für einen Angreifer die Wahrscheinlichkeit des Auftretens eines Ereignisses *vor und nach* jeder Beobachtung *gleich* ist.

Anonymität bedeutet, daß für einen Angreifer die Wahrscheinlichkeit, daß eine Instanz (z. B. eine Person) bei einem Ereignis eine bestimmte Rolle (z. B. Sender der Nachricht) wahrnimmt, *nach* jeder Beobachtung sowohl echt größer 0 als auch echt kleiner 1 ist. Perfekte Anonymität bedeutet, daß für einen Angreifer die Wahrscheinlichkeit, daß eine Instanz bei einem Ereignis eine bestimmte Rolle wahrnimmt, *vor und nach* jeder Beobachtung *gleich* ist.

1.2 Standardtechniken

Aus Platzgründen muß in diesem Beitrag darauf verzichtet werden, die Standardtechniken zur Anonymität in aller Ausführlichkeit zu behandeln.[1] Es wird jedoch im folgenden eine Systematik angegeben und jede Technik kurz erläutert.

- **Schutz des Empfängers**
  - **Verteilung (Broadcast)**: Um den Empfang einer Nachricht zu verbergen, werden Nachrichten an alle Teilnehmer verteilt. Eine Auswahl

[1] Eine Übersichtsdarstellung findet sich in Roessler, in diesem Heft.

der Nachrichten erfolgt lokal beim Empfänger. Beispiel: Rundfunkempfang.

- **Implizite Adressen**: Soll trotz Verteilung eine Punkt-zu-Punkt-Kommunikation erreicht werden, muß der Empfänger über sog. implizite Adressen adressiert werden. Implizite Adressen sind Bitketten, an denen nur der Empfänger erkennt, daß eine Nachricht für ihn bestimmt ist. Für alle anderen Teilnehmer (inkl. Angreifer) sind sie mit nichts und niemandem verkettbar. Beispiel: Ein Teilnehmer wählt sich eine Zufallszahl und teilt diese den potentiellen Sendern (z.B. in einer Chiffreanzeige) mit.

- **Schutz des Senders**
  - **Dummy Traffic**: Soll verborgen werden, ob ein Teilnehmer gerade eine Nachricht senden möchte, muß er immer senden. Solange er keine Nachrichten zu senden hat, sendet er Leernachrichten, die sich für einen Außenstehenden nicht von echten Nachrichten unterscheiden.
  - **DC-Netz**: Das DC-Netz [Chau_88] schützt den Sender einer Nachricht. So kann ein Teilnehmer eine Nachricht senden, ohne daß er als Sender erkannt wird. Der Teilnehmer ist innerhalb einer Gruppe, der sog. Anonymitätsgruppe, anonym.
- **Schutz der Kommunikationsbeziehung zwischen Teilnehmern**
  - **Mixe**: Mixe [Chau_81] schützen die Kommunikationsbeziehung, indem sie die Verkettung der Endpunkte einer Kommunikationsverbindung verhindern. Auch hier ist der Teilnehmer innerhalb einer Anonymitätsgruppe geschützt.
- **Schutz der Aufenthaltsorte von mobilen Teilnehmern**
  - **Verteilung (Broadcast)**: Werden Nachrichten in das gesamte Aufenthaltsgebiet verteilt, ist auch der Aufenthaltsort des mobilen Teilnehmers geschützt [Pfit_93].
  - **Pseudonymisierung**: Aufenthaltsdaten werden beim Netzbetreiber unter einem Pseudonym gespeichert. Die Verkettung zwischen der Identität und dem Pseudonym unterliegt der Kontrolle des Teilnehmers [KeFo_95].
  - **Mobilkommunikationsmixe**: Auch hier schützen die Mixe die Kommunikationsverbindung, allerdings wird bereits die Signalisierverbindung zwischen den Aufenthaltsdatenbanken und dem aktuellen Aufenthaltsgebiet geschützt [FeJP_96].

Weitere Informationen zu den Standardmechanismen zum Schutz des Senders, des Empfängers und der Kommunikationsbeziehung findet man z.B. in [FePf_97]. Zum Schutz der mobilen Teilnehmer siehe [Fede_98].

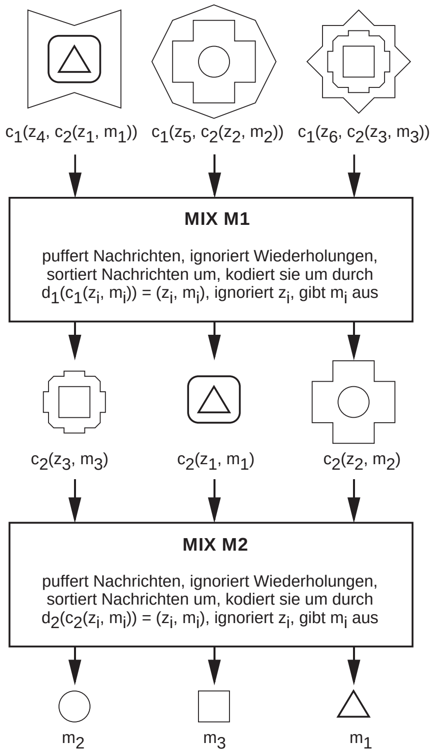


Bild 1: Umkodieren zu mixender Nachrichten

Um einen optimalen Schutz der Teilnehmer zu erreichen, müssen Sicherheitsmechanismen miteinander kombiniert werden. Bei der Verteilung persönlicher Nachrichten muß beispielsweise implizite Adressierung angewendet werden. Die Nachricht selber sollte verschlüsselt sein.

1.3 Beispiel: Mixe

Die Idee der Mixe wurde in [Chau_81] vorgestellt. Das Mix-Konzept kommt in Vermittlungsnetzen zum Einsatz. Ein Mix verbirgt die Kommunikationsbeziehung zwischen Sender und Empfänger einer Nachricht. Hierzu muß ein Mix eingehende Nachrichten speichern, bis genügend viele Nachrichten von genügend vielen Absendern vorhanden sind, ihr Aussehen verändern, d.h. sie umkodieren, und die Reihenfolge der ausgehenden Nachrichten verändern, d.h. sie umsortieren und in einem Schub ausgeben.

Um Angriffe durch Nachrichtenwiederholung zu verhindern, muß zu Beginn noch geprüft werden, ob eine eingehende Nachricht bereits gemixt wurde. Da ein Mix deterministisch umkodiert, würde eine Nachrichtenwiederholung z. B. in einem nächsten Schub zur Ausgabe der gleichen umkodierten Nachricht führen. Somit wäre eine Verkettung von Ein- und Ausgabe möglich.

Damit keine Verkettung zwischen eingehenden und ausgehenden Nachrichten über deren Länge möglich ist, sollten alle Nachrichten die gleiche Länge haben.

Eine Nachricht, die einen Mix durchläuft, ist nur innerhalb eines Schubes anonym. Deshalb muß sichergestellt sein, daß ein Angreifer nie alle Nachrichten außer einer selbst erzeugt hat, denn das käme der Deanonymisierung gleich. Arbeiten alle anderen Sender und Empfänger der in einem Schub gemixten Nachrichten zusammen, sind die Kommunikationsbeziehungen ebenfalls aufdeckbar.

Falls nicht genügend eingehende Nachrichten vorhanden sind, müssen künstliche erzeugt werden, damit die Verzögerungszeit einer Nachricht minimiert wird (Dummy Traffic).

Die Kernfunktion eines Mixes ist das Umkodieren der Nachrichten. Hierzu wird mit Hilfe eines asymmetrischen Kryptosystems jede zu mixende Nachricht mit dem privaten Schlüssel des Mixes entschlüsselt (umkodiert) und an den nächsten Mix weitergeschickt (Bild 1). Bei einem asymmetrischen Kryptosystem (z. B. RSA) haben Sender und Empfänger unterschiedliche Schlüssel zum Ver- und Entschlüsseln. Der Verschlüsselungsschlüssel (public key) ist öffentlich bekannt, d.h. jeder kann eine Nachricht für den Empfänger verschlüsseln. Den Entschlüsselungsschlüssel (private key) besitzt nur der Empfänger (hier: der Mix) und nur er kann somit seine empfangenen Nachrichten entschlüsseln.

Mehrere unabhängige Betreiber der zwischengeschalteten Mixe garantieren die Unbeobachtbarkeit der Kommunikationsbeziehungen. Solange mindestens ein Mix „gutartig“ ist (d.h. nicht mit den anderen zur Aufhebung der Anonymität kooperiert), bleibt die Kommunikationsbeziehung geschützt.

Das Mix-Konzept war in den letzten zehn Jahren Gegenstand vieler Forschungsarbeiten, zum Teil mit Grundlagencharakter, siehe z. B. [PfWa_87, PfPW_88, Pfit_90, PfPf_90], aber auch mit einer deutlichen Anwendungsorientierung, siehe z. B. [PfPW_89, Pfit_93, Cott_95, FeJP_96, FaKK_96, GoRS_96].

2 Angreifermodell

Aussagen über den erzielten Schutz können nur im Zusammenhang mit der unterstellten Stärke eines Angreifers gemacht werden. Ein Angreifermodell gibt die maximal mögliche Stärke des Angreifers an, gegen den der Schutz gerade noch gewährleistet ist.

Bezogen auf die Beobachtbarkeit im Internet und speziell im World Wide Web wollen wir Schutz vor einem Angreifer erreichen, der auf allen Leitungen alle Kommunikation abhören kann.[2] Verkehrsanalysen sind für ihn möglich. Auch Insider (Nutzer/Beobachter im Intranet) werden als Angreifer betrachtet.

Eine Konsequenz für den Schutz vor Beobachtbarkeit ist, daß bei diesem Angreifermodell die Nutzung von http-Proxies (z. B. Anonymizer, siehe Abschnitt 3.1) zur Verschleierung von Webanfragen nicht ausreicht (Bild 2), da der Angreifer auch im Intranet verbreitet sein kann.

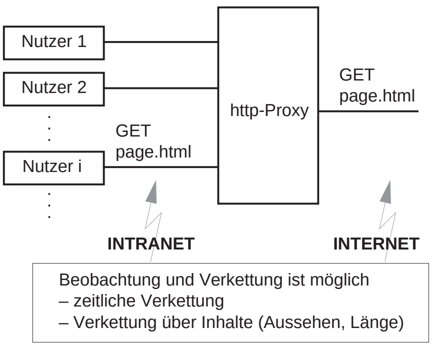


Bild 2: Verbreitung des Angreifers

[2] Zwar ist dies eine strenge Annahme, wenn aber eine Kommunikationsbeziehung auch unter diesen Bedingungen anonym ist, kann auch ein schwächerer Angreifer die Anonymität nicht aufheben.

3 Bewertung der „neuen“ Techniken

Durch die starke Verbreitung des Internet und die zunehmende Abwicklung persönlicher Kommunikation gewinnen die Entwicklung und insbesondere die Implementierung von Verfahren zur Unbeobachtbarkeit und Anonymität große Bedeutung.

In folgenden sollen drei vergleichsweise neue Ansätze, die den Schutz vor Beobachtung insbesondere im Internet gewährleisten sollen, auf ihre Grenzen untersucht werden.

- Anonymizer (www.anonymizer.com),
- Crowds (www.research.att.com/projects/crowds),
- Onion Routing (www.onion-router.net).

Obwohl alle drei Ansätze annähernd das gleiche Ziel verfolgen, unterscheiden sie sich insbesondere in dem erreichbaren Schutz, d. h. die Angreifermodelle der Verfahren sind unterschiedlich stark.

Ziel der Verfahren ist, gegenüber dem Server und teilweise auch gegenüber Beobachtern, die Verkehrsanalysen durchführen, zu verbergen, wer welche Webseiten aufruft (Client-Anonymität).[3]

Die oben genannten Verfahren werden im folgenden kurz beschrieben und dann insbesondere im Hinblick auf den erreichten Schutz miteinander verglichen.

3.1 Anonymizer

Der „Anonymizer“ ist ein Proxydienst. Über ein Webinterface (einfacher Aufruf einer Webseite, in die eine URL eingegeben werden muß) kann der Dienst genutzt werden.

Der Nutzer muß dem Betreiber des Anonymizer vertrauen, daß er keine Verkehrs- und Interessensdaten sammelt. Theoretisch können mehrere Anonymizer hintereinander geschaltet (kaskadiert) werden. Dann weiß nur noch der erste Anonymizer direkt, wer (genauer: welche IP-Adresse) den Dienst nutzt.

Technisch gesehen arbeitet der Anonymizer wie ein herkömmlicher http-(Web-) Proxy, jedoch mit dem Unterschied, daß der Anonymizer alle potentiell personenbezogenen Informationen (z. B. Cookies) in den Headern der Webanfragen entfernt. Verschlüsselung wird beim Anonymizer nicht verwendet.

[3] Zur Anonymität des Servers siehe Demuth, Rieke, in diesem Heft.

Bezüglich eines Angreifers, der alle Kommunikation im Netz abhören kann bzw. Verkehrsanalysen durchführt, ist der Anonymizer nicht sicher. Somit gelten sinngemäß die Aussagen zu http-Proxys aus Abschnitt 2. Das bedeutet, die Anfragen können über das „Aussehen“ der Nachrichten und deren Länge sowie die zeitlichen Korrelationen der ein- und ausgehenden Nachrichten verkettet werden. Außerdem kann der Angreifer sofort alle Inhalte mitlesen, da keine Verschlüsselung verwendet wird.

Die Bemerkungen zum Anonymizer gelten auch für andere einfache Proxys, z.B. den Lucent Personalized Web Assistant (LPWA, siehe www.bell-labs.com/project/lpwa/overview.html). Der Nutzer muß dem System vollständig vertrauen, da sämtliche Zugriffe und Daten von einer zentralen Instanz verwaltet werden.

3.2 Crowds

Crowds „versteckt“ die Webanfragen eines Teilnehmers in denen der anderen Crowds-Dienstnutzer. Um am Dienst teilzunehmen, meldet sich der Nutzer bei einer zentralen Stelle, dem sogenannten Blender, an.

Auf dem lokalen Rechner hat jeder am Dienst teilnehmende Nutzer ein Programm installiert, den sogenannten Jondo. Die Idee von Crowds ist, daß eine Webanfrage nicht direkt an den Server gestellt wird, sondern vorher mehrere Jondos anderer Teilnehmer durchläuft. In jedem Jondo wird eine Anfrage zufällig entweder zu einem weiteren Jondo geschickt oder direkt an den Server.

Ein negativer Aspekt von Crowds ist, daß ein Teilnehmer fälschlicherweise für den Absender einer Anfrage gehalten werden kann. Ein Vorteil ist jedoch, daß ein Teilnehmer stets abstreiten kann, der Initiator einer Anfrage gewesen zu sein.

Ein Angreifer, der die Kommunikationsinhalte unmittelbar mitlesen will, hat bei Crowds im Gegensatz zum Anonymizer keine Möglichkeit, da die Inhaltsdaten zwischen den Jondos mit einem symmetrischen Kryptosystem[4] verschlüsselt sind.

Eine Verkettung über die Länge der (verschlüsselten) Nachrichteninhalte und damit die Beobachtung ist jedoch nach wie vor möglich, wenn der Angreifer Verkehrsanalysen durchführt. Gegen Angriffe über die zeitliche Verkettung von eingehenden Nachrichten eines Jondos und deren Ausgabe wurden keine Schutzmaßnahmen vorgesehen.

3.3 Onion Routing

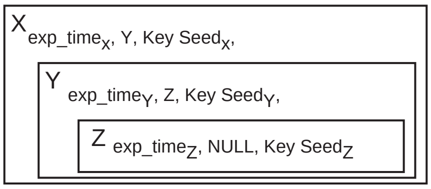
Onion Routing beschränkt sich nicht auf Webzugriffe. Es ist ebenfalls für Filetransfer (ftp), Remote Login und andere verbindungsorientierte Dienste nutzbar. Es arbeitet als Proxydienst mit einem sog. Initiator-Proxy auf der Nutzerseite und einem Responder-Proxy auf der dem Internet „zugewandten“ Seite. Zwischen Initiator- und Responder-Proxy sind mehrere Onion-Router geschaltet.

Onion Routing soll folgendes Schutzziel erfüllen: Ein Angreifer, der alle Kommunikation im Netz abhören kann, soll nicht in der Lage sein, ein- und ausgehende Nachrichten eines Onion-Routers miteinander zu verketten. Dieses Angreifermodell entspricht praktisch dem von David Chaum aus [Chau_81]. Die technische Lösung für das Onion Routing ähnelt Chaums Idee der Mixe sehr stark (siehe Abschnitt 1.3)

Chaum hatte damals für seine Mixe die Elektronische Post (E-Mail) als Dienst gewählt. Sie hat den Vorteil, nicht von Echtzeitanforderungen und Verbindungen abhängig zu sein. Für das World Wide Web benötigt man jedoch eine zumutbare Verzögerungszeit, bis die Antwort auf eine Anfrage im Browser zu sehen ist. Daher waren Modifikationen des Gundkonzeptes notwendig, die zum Teil zu Lasten des erreichten Schutzes gehen.

Beim Onion Routing wird zunächst über eine Kanalaufbaunachricht (`create`) eine Onion (siehe Bild 3, Aufbau für drei Onion-Router X, Y, Z) gesendet. Die Onion enthält eine Zeitangabe (`exp_time`), die angibt, wann eine Onion verfällt. Die Zeitangabe dient der Abwehr von Nachrichtenwiederholungen. Solange `exp_time` noch nicht abgelaufen ist, speichert der Onion-Router die Onion und testet auf Nachrichtenwiederholung. Weiterhin enthält die Onion die Adresse des nächsten Onion-Routers sowie Schlüsselmaterial, das für die nachfolgende Etablierung des „anonymen Kanals“ verwendet wird.

Bei Ihrem Lauf durch das Netz wird die Onion Schritt für Schritt abgebaut, d. h. im jeweiligen Onion-Router entschlüsselt, und gleichzeitig der anonyme Kanal aufgebaut. Hierzu merkt sich jeder Onion-Router, woher er eine Onion erhalten, wohin er die verbleibende Onion geschickt hat und zusätzlich ein Kennzeichen, die sogenannte Pfad-ID. Empfängt ein Onion-Router Daten für eine bestimmte ID, so verschlüsselt er die erhaltenen Daten mit einem symmetrischen Kryptosystem, dessen Schlüssel er aus dem Schlüsselmaterial (Key Seed) der Onion gewonnen hat.



Jede „Schale“ ist mit dem public key des nächsten Onion-Routers verschlüsselt.

Bild 3: Aufbau einer Onion

Dummy Traffic wird nur zwischen den Onion-Routern erzeugt und bietet somit bei geringer Auslastung des Dienstes keinen (bzw. nur geringen) Schutz gegen Beobachtung, da die Enden eines Kommunikationskanals allein über die ausgetauschte Datenmenge verkettet werden können.

Da die über die anonyme Verbindung laufenden Daten eine beliebige Länge haben können, ist eine Verkettung über die Länge der über den anonymen Kanal gesendeten Nachrichten möglich.

	Zeitliche Verkettung	Verkettung über Inhalt
Anonymizer	keine Vorkehrungen dagegen	keine Vorkehrungen, lediglich Headerinformationen werden entfernt
Crowds	keine Vorkehrungen, aber wenigstens Zusammenfassung von Anfragen in Jondos	keine Vorkehrungen, aber wenigstens sind Inhalte verschlüsselt
Onion Routing	schwache Vorkehrungen, lediglich Dummy Traffic zwischen Onion-Routern	für Kanalaufbau keine Verkettung, für Datenaustausch jedoch Verkettung über Nachrichtenlänge möglich

Tabelle 1: Vergleich der Verfahren

4 Fazit und Ausblick

In Tabelle 1 werden die genannten Verfahren noch einmal nach dem erreichten Schutz gegenüber Verkehrsanalysen zusammengefaßt.

Sofern überhaupt eine Metrik über dem erreichten Schutz der drei vorgestellten Verfahren sinnvoll ist, kann man feststellen, daß Crowds gegen stärkere Angriffe schützt als Anonymizer, und Onion Routing stärker schützt als Crowds.

David Chaum präsentierte 1981 in seinem Mixe-Artikel [Chau_81] eine Lösung, bei der aus der Sicht der Teilnehmer lediglich eine von *n* Parteien (n>1) vertrauenswürdig sein muß, damit der Teilnehmer geschützt ist. Im Grenzfall ist jeder Teilnehmer eine der *n* Parteien, und damit nehmen die Nutzer ihren Schutz in die eigenen Hände. Dies geschieht z. B. bei Crowds.

Keines der hier vorgestellten Verfahren bietet perfekte Anonymität. Die Verfahren erfüllen nicht einmal das diesen Untersuchungen zugrunde gelegte Angreifermodell („Alle Kommunikation ist überwachbar.“)

Erste Ansätze zu einer Lösung, die auch noch gegen das starke Angreifermodell schützt, wurden in [FeMa_98] und auf der Cebit 1998 vorgestellt. Dort soll auch die zeitliche Verkettung und Verkettung über die Nachrichtenlänge nicht mehr möglich sein. Angewendet wird eine Modifikation der Zeitscheiben der ISDN-Mixe (siehe [PfPW_89]). Beim Zeitscheibenmodell wird ein anonymer Kanal aufgebaut, für eine festgelegte Zeit bzw. Nachrichtenmenge genutzt und anschließend wieder abgebaut. Dieser Prozeß wiederholt sich ständig und ein Teilnehmer ist innerhalb der während der Zeitscheibe gemeinsam verarbeiteten Nachrichten anonym, da alle die gleiche Zeit bzw. Nachrichtenlänge verwenden. Zwi-

[4] Bei einem symmetrischen Kryptosystem besitzen sowohl Sender als auch Empfänger den gleichen Schlüssel zum Ver- und Entschlüsseln.

schen den Zeitscheiben besteht keine Verkettung.

Literatur

Chau_81 David Chaum: *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*. Communications of the ACM 24/2 (1981) 84-88.

Chau_88 David Chaum: *The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability*. Journal of Cryptology 1/1 (1988) 65-75.

Cott_95 Lance Cottrel: *Mixmaster & Remailer Attacks*. http://www.obscura.com/~loki/remailer-essay.html.

FaKK_96 Andreas Fasbender, Dogan Kesdogan, Olaf Kubitz: *Scalable Security: Protection of Location Information in Mobile IP*. Proceedings of IEEE VTS 46th Vehicular Technology Conference, Atlanta, USA, April 28 - Mai 1, 1996 .

Fede_98 Hannes Federrath: *Sicherheit mobiler Kommunikation*. erscheint bei Vieweg.

FeJP_96 Hannes Federrath, Anja Jerichow, Andreas Pfitzmann: *Mixes in mobile communication systems: Location management with privacy*. in: R. Anderson (Hrsg.): Information Hiding, LNCS 1174, Springer-Verlag, Berlin 1996, 121-135.

FeMa_98 Hannes Federrath, Kai Martius: *Mehrseitig sicherer Web-Zugriff*. KES Zeitschrift für Kommunikations- und EDV-Sicherheit 14/4 (1998) 10-12.

FePf_97 Hannes Federrath, Andreas Pfitzmann: *Bausteine zur Realisierung mehrseitiger Sicherheit*. in: Günter Müller, Andreas Pfitzmann (Hrsg.): Mehrseitige Sicherheit in der Kommunikationstechnik, Addison-Wesley-Longman 1997, 83-104.

GoRS_96 David M. Goldschlag, Michael G. Reed, Paul F. Syverson: *Hiding Routing Information*. in: R. Anderson (Hrsg.), Information Hiding, LNCS 1174, Springer-Verlag, Berlin 1996, 137-150.

KeFo_95 Dogan Kesdogan, Xavier Fouletier: *Secure Location Information Management in Cellular Radio Systems*. Proc. IEEE Wireless Communication System Symposium 95, Long Island (1995), 35-46.

Pfit_90 Andreas Pfitzmann: *Diensteintegrierende Kommunikationsnetze mit teilnehmerüberprüfbarem Datenschutz*. IFB 234, Springer-Verlag, Heidelberg 1990.

Pfit_93 Andreas Pfitzmann: *Technischer Datenschutz in öffentlichen Funknetzen*. Datenschutz und Datensicherung DuD 17/8 (1993) 451-463.

PfPf_90 Birgit Pfitzmann, Andreas Pfitzmann: *How to Break the Direct RSA-Implementation of MIXes*. Eurocrypt '89, LNCS 434, Springer-Verlag, Berlin 1990, 373-381.

PfPW_88 Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: *Datenschutz garantierende offene Kommunikationsnetze*. Informatik-Spektrum 11/3 (1988) 118-142.

PfPW_89 Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: *Telefon-MIXe: Schutz der Vermittlungsdaten für zwei 64-kbit/s-Duplexkanäle über den (2x64 + 16)-kbit/s-Teilnehmeranschluß*. Datenschutz und Datensicherung DuD 13/12 (1989) 605-622.

PfWa_87 Andreas Pfitzmann, Michael Waidner: *Networks without user observability*. Computers & Security 6/2 (1987) 158-166.