JANUS: Server-Anonymität im World Wide Web

Andreas Rieke · Thomas Demuth

FernUniversität Hagen
Fachgebiet Kommunikationssysteme
Feithstr. 142, 58084 Hagen
{andreas.rieke,thomas.demuth}@fernuni-hagen.de

Zusammenfassung

In den vergangenen Jahren sind viele Verfahren entwickelt worden, um den Inhalt von Nachrichten vor Abhörern zu schützen. Oft ist jedoch nicht nur der Inhalt, sondern auch die Adresse und Identität des Absenders und/oder Empfängers für Angreifer von Interesse. Aus diesem Grund haben sich mehrere Projekte mit dem Ziel beschäftigt, Anonymität im Fall von Email zu realisieren und zu garantieren.

Heutzutage befassen sich einige Vorhaben mit verschiedenen Möglichkeiten, um Client-Anonymität im World Wide Web (WWW) zu garantieren, allerdings gibt es bis heute keinen Dienst, der Server-Anonymität leistet. Basierend auf Chaums Lösung für unverfolgbare Email führen wir eine neue Lösung für Server-Anonymität ein, die anonymes Veröffentlichen im WWW ermöglicht, und präsentieren einen Prototypen namens JANUS, der sowohl Client- als auch Server-Anonymität garantiert.

1 Einleitung

Nachdem das Verschlüsseln des Inhaltes einer Nachricht Thema in verschiedensten Forschungsprojekten war, wird die Anonymität in Kommunikationssystemen heutzutage ein immer wichtigerer Faktor. Anonymität in Kommunikationssystemen wurde beispielsweise in [PW87] wie folgt klassifiziert:

- Empfängeranonymität: Der Empfänger einer Nachricht bleibt anonym. Eine einfache, aber unpraktische Methode liegt darin, eine Nachricht an eine große Anzahl von Empfängern zu schicken.
- Unbeobachtbarkeit der Kommunikationsbeziehung: Die Unbeobachtbarkeit einer Kommunikationsbeziehung liegt vor, wenn die Beziehung zwischen Absender und Empfänger einer Nachricht für Dritte nicht nachvollziehbar ist.
- Senderanonymität: Unter Verwendung von Senderanonymität kann der Empfänger einer Nachricht nicht feststellen, wer diese Nachricht abgeschickt hat.

Anonymität für die Kommunikation per Email wurde bereits in mehreren Projekten untersucht. Das exponentielle Wachstum des Internet wurde jedoch nicht durch Email, sondern

durch das World Wide Web (WWW) ausgelöst, das in erster Linie auf dem Hypertext Transfer Protocol (HTTP) basiert.

Da HTTP und verwandte Protokolle nicht mit den Begriffen Sender und Empfänger arbeiten, führen wir im folgenden die Notation von Client- und Server-Anonymität ein. Nur einige wenige Projekte haben sich mit Anonymität im WWW befaßt und die meisten hiervon beschränken sich auf Client-Anonymität. Einige Ansätze haben ebenfalls die Unbeobachtbarkeit der Kommunikationsbeziehung einbezogen, aber Server-Anonymität wurde im Internet noch nicht realisiert.

In diesem Papier analysieren wir Server-Anonymität im WWW und präsentieren einen ersten Prototypen namens JANUS (Justly Anonymizing Numerous URLs Systematically), der sowohl Client- als auch Server-Anonymität unterstützt.

Dieses Papier ist wie folgt strukturiert: Nach dieser Einleitung wird in Kapitel 2 ein grundlegendes Konzept zur Anonymität – das Mixe-Konzept von David Chaum – beschrieben. Kapitel 3 beschreibt anschließend bekannte Ansätze zur Client-Anonymität im World Wide Web. Kapitel 4 führt die Idee der Server-Anonymität ein; es werden Beispiele angeführt, in denen Server-Anonymität benötigt wird und es wird kurz beschrieben, wie Server-Anonymität realisiert werden kann. Im folgenden Kapitel 5 wird unser Prototyp beschrieben. Einige bekannte Probleme werden in Kapitel 6 diskutiert, bevor die wesentlichen Ergebnisse dieses Papiers in Kapitel 7 zusammengefaßt werden.

2 Mixe

1981 veröffentlichte David Chaum ein Konzept [Cha81], das unverfolgbare Email, Rückadressen und digitale Signaturen beinhaltet. Nach dem Ansatz bestimmt der Absender einer Email eine Route zum Empfänger, die über mehrere Zwischenstationen, die sogenannten Mixe, verläuft. Jeder dieser Mixe verfügt über ein asymmetrisches Verschlüsselungsverfahren mit der Eigenschaft

$$d(e(x)) = x = e(d(x)),$$

wobei $e(x)$ die Verschlüsselungsfunktion und $d(x)$ die Entschlüsselungsfunktion darstellt. Wir werden im folgenden annehmen, daß die Verschlüsselungsfunktionen aller beteiligten Parteien in authentischer Form verfügbar sind.

Bevor der Absender A eine Email abschickt, verschlüsselt er den Inhalt mit der Verschlüsselungsfunktion des Empfängers. Die Adresse des Empfängers wird an den verschlüsselten Inhalt angehängt, um beides wiederum mit der Verschlüsselungsfunktion des letzten Mixes auf der Route zu verschlüsseln.

Im nächsten Schritt wird die Adresse des letzten Mixes zur Nachricht hinzugefügt und beides wird mit der öffentlichen Verschlüsselungsfunktion des vorherigen Mixes verschlüsselt. Diese Prozedur wird wiederholt, bis der erste Mix der Route erreicht wird. Nachdem der Inhalt für diesen Mix verschlüsselt wurde, wird die Nachricht an ihn versandt.

Jeder Mix, der eine Nachricht erhält, entschlüsselt diese mit seiner geheimen Entschlüsselungsfunktion. Der entschlüsselten Nachricht entnimmt er die Adresse und die Nachricht für den nächsten Empfänger und leitet die Daten an diesen weiter.

Als Beispiel wird die Nachricht M an die Adresse A_B über die Mixe mit den Adressen A_1 und A_2 gesandt, die über die Verschlüsselungsfunktionen $e_1(x)$ und $e_2(x)$ verfügen. $e_B(x)$ ist die Verschlüsselungsfunktion des Empfängers. Mit "," als Symbol für das Aneinanderhängen von Daten wird

$$e_2(e_1(e_B(M), A_B), A_1)$$

als Nachricht zum ersten Mix A_2 geschickt.

Falls x einfach mittels $e(x)$ verschlüsselt würde, könnte jedermann testen, ob $y = x$, indem $e(y) = e(x)$ überprüft wird. Um diese Bedrohung auszuschließen, wird ein langer String von Zufallsbits R zu x hinzugefügt, bevor beides verschlüsselt wird. Mit dieser Verbesserung wird

$$e_2(e_1(e_B(M, R_0), A_B, R_1), A_1, R_2)$$

als Nachricht zum ersten Mix A_2 geschickt.

Mit diesem Ansatz kann die Anonymität des Absenders garantiert werden; indem der Ansatz umgekehrt wird, wird Empfängeranonymität erreicht: Um dem Empfänger B einer anonymen Email zu ermöglichen, diese zu beantworten, ohne die Anonymität aufzudecken, muß der Absender der ursprünglichen Nachricht A seine eigene, verschlüsselte Adresse in die Email einfügen. Daher muß der Absender auch eine Route vom Empfänger zurück zu sich selbst finden, die wiederum über mehrere Mixe verläuft.

Beginnend mit dem Mix, der ihm selbst am nächsten liegt, verschlüsselt er seine eigene Adresse mit der Verschlüsselungsfunktion des Mixes. Diese Adresse wird dann mit der Verschlüsselungsfunktion des folgenden Mixes verschlüsselt, und das Ganze wird wiederholt, bis der letzte Mix erreicht ist. Bei n Mixen liegt dann folgendes vor:

$$e_1(R_1, e_2(R_2, \cdots, e_{n-1}(R_{n-1}, e_n(R_n, A_A))\cdots))$$

Mit der Funktion $r_i(x)$, die mit R_i als geheimem Schlüssel arbeitet, verschlüsseln B und alle Mixe den Inhalt der Antwort, da R_i nur dem jeweiligen Mix und A bekannt ist. Daher sendet B

$$r_0(M)$$

und A empfängt

$$r_n(r_{n-1}(\cdots r_2(r_1(r_0(M)))\cdots)).$$

3 Client-Anonymität

Client-Anonymität bedeutet, daß in einem Kommunikationsverhältnis nach dem Client-Server-Modell alle Informationen über den Client verborgen werden, d. h. der Client bleibt anonym.

Es gibt zwei Wege, auf denen ein Server oder ein Dritter Informationen über einen Client beziehen kann:

1. Im Internet-Protokoll (IP) enthält jedes Paket seine Quell- und Zieladresse. Bei einer direkten Verbindung kann ein Server oder ein Angreifer daher die IP-Adresse des Client bestimmen.
2. Die Daten, die zwischen Client und Server ausgetauscht werden, enthalten Verwaltungsinformationen, über die auf den Client geschlossen werden kann. Im Fall von HTTP, das oft benutzt wird, kann die Client-Software (z. B. der Netscape Communicator oder der Microsoft Internet Explorer) Daten wie die Email-Adresse des Client oder die Adresse der letzten Seite an den Server weitergeben.

In den folgenden Kapiteln werden Ansätze, die Client-Anonymität garantieren, beschrieben. Jeder dieser Ansätze wurde wenigstens einmal implementiert.

3.1 Proxies

Proxies erhalten Anfragen von Clients und leiten diese an Server weiter. In der ursprünglichen Bedeutung speichern sie Informationen zwischen und sorgen dadurch für einen schnellen Zugriff auf häufig benötigte Informationen und vermeiden unnötigen Verkehr auf dem Netz.

Im Gegensatz zu gewöhnlichen Proxies anonymisieren Proxies im Sinne dieses Papiers sowohl Anfragen als auch Antworten (Verweise werden beispielsweise so geändert, daß die entsprechenden Inhalte ebenfalls anonym geladen werden). Elemente, die die Anonymität gefährden, werden dabei entfernt.

Nachteile dieses Ansatzes sind:

- Der Anwender muß dem Proxy trauen.
- Obwohl die Verkettung von Proxies möglich ist, bringt sie keine Vorteile.

Der Anonymizer (http://www.anonymizer.com/) war einer der ersten Anonymitätsdienste, der im Internet basierend auf dem Proxy-Ansatz angeboten wurde. Um die Wichtigkeit anonymer Verbindungen zu demonstrieren, bietet der Anonymizer eine Seite an, auf der alle Informationen über den Nutzer dargestellt werden. Ein Beispiel ist:

> Your name is probably Andreas Rieke, and you can be reached at rieke@corona.fernuni-hagen.de. .... Your connection provider is located in Germany (Federal Republic of). Your computer is a Unix box running SunOS 5.5.1 sun4u. Your Internet browser is Netscape. You are coming from corona.fernuni-hagen.de. You just visited the Anonymizer Home Page.

Ein anderer, umfassenderer Ansatz, der Lucent Personalized Web Assistant ([BGG+98], [GGMM97], http://lpwa.com:8000/), wurde in den Bell Labs bzw. bei Lucent Technologies entwickelt. Neben Client-Anonymität bietet dieser ebenfalls Vertraulichkeit und Zugangsschutz.

3.2 Crowds

Crowds arbeitet mit einer großen, geographisch verteilten Gruppe, die kollektiv Anfragen im Auftrag ihrer Mitglieder stellt. Anstatt ein Dokument direkt von einem Web-Server abzurufen, leitet ein Mitglied der Gruppe die Anfrage an ein zufällig ausgewähltes anderes Mitglied weiter. Dieses Mitglied schickt die Anfrage entweder direkt an den Web-Server oder an wiederum ein anderes, zufällig ausgewähltes Mitglied. Jedes Mitglied entscheidet zufällig anhand einer Wahrscheinlichkeit p_f, ob es die Anfrage an ein anderes Mitglied oder an den Web-Server weiterleitet.

Web-Server können die ursprüngliche Quelle der Anfrage nicht feststellen, da diese von jedem Mitglied der Gruppe kommen kann. Selbst eine Untergruppe von korrupten Mitgliedern kann nicht feststellen, woher die ursprüngliche Anfrage kam, da auch das Mitglied aus der Untergruppe, das die Anfrage zuerst bekommt, nicht mit Sicherheit feststellen kann, woher die Anfrage tatsächlich kommt.

Da der Verkehr zwischen den Mitgliedern verschlüsselt ist, kann ein lokaler Angreifer weder den Inhalt der Anfrage noch den tatsächlichen Empfänger feststellen. Trotzdem hat das Crowds-System einige Nachteile:

- Ein Mitglied der Gruppe kann verdächtigt werden, einen Zugriff durchgeführt zu haben, der im Namen eines anderen Gruppenmitglieds durchgeführt wurde.
- Die Wahrscheinlichkeit p_f, eine Anfrage an ein anderes Mitglied weiterzuleiten, muß geeignet festgelegt werden:
  - Im Fall einer korrupten Untergruppe kann das erste Mitglied in einer Anfragekette annehmen, daß das Mitglied, das diese Anfrage durchgeführt hat, die ursprüngliche Quelle ist. Je niedriger p_f gewählt wird, desto höher ist die Wahrscheinlichkeit, daß diese Annahme richtig ist.
  - Falls p_f zu hoch angesetzt wird, steigt die Anzahl der Folgeanfragen, wodurch die Leistungsfähigkeit des Systems, die Geschwindigkeit und die Zuverlässigkeit verringert werden.
- Die Anzahl der Gruppenmitglieder n sollte genügend groß sein.
- Es ist unwahrscheinlich, daß alle Gruppenmitglieder vertrauenswürdig sind. Da alle Zugriffe für die betroffenen Mitglieder sichtbar sind, sind diese in der Lage, sensible Informationen wie Paßwörter oder Kreditkarteninformationen zu sammeln.

Das Crowds Projekt ([RR97], http://www.research.att.com/projects/crowds/) wurde in den AT&T Labs durchgeführt.

3.3 Onion Routing

Im Onion (Zwiebel) Routing Ansatz wird eine Anfrage beliebiger Art zunächst an einen Client Proxy, der vom Client betrieben wird, geschickt. Dieser Client Proxy berechnet eine Route über einen oder mehrere Hauptproxies. Außerdem verschlüsselt er die Anfrage und die Adres-

se des Servers mit dem öffentlichen Schlüssel des letzten Hauptproxies auf der Route. Dies ist der Kern der Zwiebel.

Die Schichten der Zwiebel werden entsprechend der anderen Hauptproxies berechnet. Für jeden werden Adresse und Inhalt mit den entsprechenden öffentlichen Schlüssel des vorherigen Hauptproxies verschlüsselt. Die so verschlüsselten Daten werden dann an den ersten Hauptproxy auf der Route weitergeleitet, der diese mit seinem geheimen Schlüssel entschlüsselt. Dadurch erhält er die Adresse des nächsten Hauptproxies und leitet die Anfrage wiederum weiter, bis diese beim Server eintrifft.

Der Onion Routing Ansatz bringt die folgenden Nachteile mit sich:

- Da der komplette Inhalt mit asymmetrischen Verfahren ver- und entschlüsselt wird, ist der Ansatz sehr zeitaufwendig.
- Wenn Firewalls eingesetzt werden, wird ein Hauptproxy auf dem Firewallrechner erforderlich.

Eine detaillierte Beschreibung des Ansatzes kann unter http://www.onion-router.net/ und in [SRG97] und [RSG98] gefunden werden.

4 Server-Anonymität

Das Ziel der Server-Anonymität liegt darin, die Adresse und damit die Identität des Servers vor dem Client zu verheimlichen. Trotzdem muß der Client in der Lage sein, eine Verbindung zum Server aufzubauen.

Im weiteren Verlauf dieses Papiers werden wir uns sowohl mit Client- als auch mit Server-Anonymität befassen. In diesem Fall kennt weder der Client noch der Server die Adresse des jeweils anderen.

4.1 Warum Server-Anonymität?

Die folgenden Beispiele beantworten diese Frage:

1. Die Begutachtung von Forschungspapieren für wissenschaftliche Konferenzen wird oft anonym durchgeführt, d. h. weder der Autor noch der Gutachter kennen die Identität des jeweils anderen. Aus diesem Grund werden die Papiere über eine vertrauenswürdige dritte Instanz weitergeleitet, ohne daß der Name des Autors oder der des Gutachters verraten wird.

   Oft wollen Autoren Verweise auf ihre eigene Arbeit einfügen, dürfen jedoch keine offensichtlichen Referenzen angeben. Eine Lösung für dieses Problem liegt darin, die entsprechenden Papiere anonym im Internet zu publizieren, ohne dabei die eigene Identität bekanntzugeben.

2. Eine elektronische Zeitung möchte Anzeigen ihrer Leser veröffentlichen. Da die Auftraggeber oft anonym auftreten wollen (Chiffreanzeigen), wird den Anzeigen eine Nummer zugeordnet, anhand derer die Zeitung die Antworten an den Auftraggeber weiterleitet.

In der elektronischen Version kann der Auftraggeber die verschlüsselte Adresse seiner elektronischen Anzeige an die Zeitung senden, die diese Adresse einfach veröffentlicht. In diesem Fall kennt nicht einmal die Zeitung die wahre Identität des Auftraggebers, obwohl der Inhalt der Anzeige auf dem WWW-Server des Inserierenden liegen kann.

3. Im jugoslawischen Bürgerkrieg war das Internet eine der wenigen Möglichkeiten für Bürgerrechtsgruppen, mit dem Rest der Welt zu kommunizieren. Die Gruppen konnten über die Ungerechtigkeit ihrer Regierung per Email und News berichten, nicht jedoch im WWW, da die Gefahr der Rückverfolgung bestand. In diesem Fall und in anderen totalitären Systemen kann Server-Anonymität die Anonymität der Meinung wenigstens im Internet garantieren und damit Verfolgung verhindern, wenn die Meinungsfreiheit nicht gewährleistet ist.

4.2 Wie wird Server-Anonymität realisiert?

Basierend auf Chaums Idee gibt es zahlreiche Punkte, die bei der Realisierung von Server-Anonymität im Internet zu beachten sind. Einer der wichtigsten ist das Transportprotokoll. Obwohl HTTP heutzutage das bekannteste Transportprotokoll ist, können andere, verwandte Protokolle wie Gopher oder FTP ähnlich behandelt werden.

Das Transportprotokoll enthält nicht nur Metainformationen über den Inhalt, sondern auch Adreßinformationen (z. B. die URL (uniform resource locator) der zu ladenden Seite) oder andere Informationen über Client oder Server (z. B. die Softwareversion oder Informationen über das verwendete Betriebssystem), die die Anonymität gefährden.

Der nächste Punkt betrifft den übertragenen Inhalt. Aus der Sicht eines Mixes wird nicht nur Inhalt transportiert, sondern auch Adreßinformationen, da der Inhalt, z. B. HTML-Seiten, Referenzen auf andere Seiten enthalten kann. Es ist wichtig, festzustellen, welcher Teil der Nachricht wirklichen Inhalt und welcher Adreßinformationen enthält, damit beides entsprechend behandelt werden kann.

5 JANUS: Der Prototyp

Unser Prototyp wurde nach JANUS, dem römischen Gott der Torbögen und Durchgänge benannt, der zwei Gesichter hatte. Der Prototyp, dessen Funktionsweise in Abb. 1 gezeigt ist, gestattet lediglich Zugriff auf öffentliche Seiten und verschlüsselt nur die Adreßinformationen, nicht jedoch den übertragenen Inhalt. Obwohl dies Angreifern ermöglicht, Informationen über den Inhalt zu bekommen, hat es die folgenden Vorteile:

- Wir sind in der Lage, Mißbrauch zu erkennen und zu verhindern. Wenn der Inhalt verschlüsselt wäre, könnten wir nicht einmal eine HTML-Seite von einer Grafik unterscheiden.
- Wegen weniger zu ver- bzw. entschlüsselnden Daten bietet diese Lösung eine höhere Geschwindigkeit.

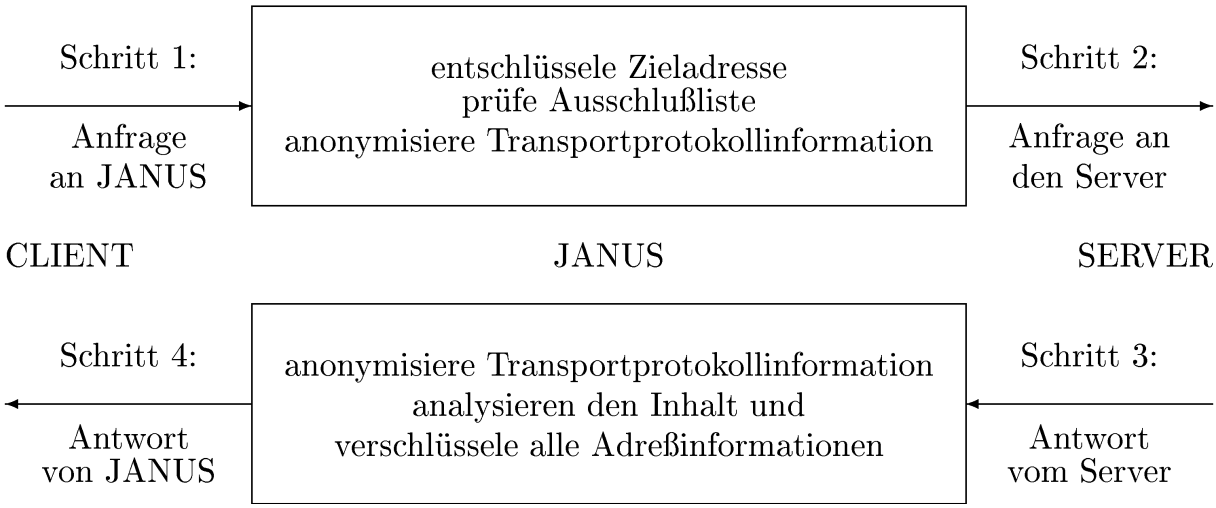

Abb. 1: Funktionsweise von JANUS

Der Prototyp ist unter http://janus.fernuni-hagen.de/ und https://janus.fernuni-hagen.de/ erreichbar. Die zweite Adresse bietet verbindungsorientierte Sicherheit auf der Verbindung vom Client zum JANUS mit Hilfe des SSL-Protokolls (Secure Socket Layer). JANUS läuft auf einer Ultra Sparc mit Perl 5.004_03, libwwwperl 5.14 und Cryptix 1.16.

5.1 Angriffsmodell

Wegen der fehlenden Verschlüsselung der übertragenen Inhalte unterscheidet sich das Angriffsmodell unseres Prototyps von dem von Chaum. Im ursprünglichen Mix-Konzept kann ein Angreifer alle Leitungen abhören und alle bis auf einen Mix übernehmen, ohne zu einer gegebenen Nachricht den Zusammenhang zwischen Absender und Empfänger finden zu können.

Im Fall des JANUS-Prototyps definieren wir zwei Vertrauensbereiche: den Client- und den Serverbereich. Jeder dieser Bereiche enthält den Client bzw. den Server, die Verbindung vom Client/Server zum nächsten JANUS und diesen JANUS-Server.

Die Verbindung von Client und Server wird offensichtlich für jeden, der in beiden Bereichen Zugang hat. Zum Beispiel erhält der Client Kenntnis von der Adresse des Servers, wenn er die Verbindung vom letzten JANUS zum Server abhört. Aus diesem Grund ist es nicht sinnvoll, mehr als zwei JANUS-Server zu benutzen, da weitere JANUS-Server die Sicherheit des Systems nicht erhöhen.

5.2 Übertragungsprotokolle

JANUS unterstützt zur Zeit HTTP/1.0 [BLFF96] und HTTPS (mit 128-Bit-Verschlüsselung) als Transportprotokoll zwischen Clients und JANUS, zwischen JANUS und dem Server wird HTTP, FTP und Gopher unterstützt. Im folgenden beschränken wir uns auf HTTP, da dies das am meisten verwendete Protokoll ist.

Neben der Anfrage- bzw. Statuszeile besteht HTTP aus mehreren Feldern, die Verwaltungsinformationen enthalten. JANUS übermittelt nur diejenigen Felder, die die Anonymität von Client und Server nicht gefährden. Aus diesem Grund werden Cookies u. a. entfernt.

5.3 Übertragener Inhalt

Der Inhalt ist nur dann von Interesse, wenn er Adreßinformationen enthält; andernfalls wird er unverändert weitergegeben. Im Fall von HTML [Rag97] werden einige Tags (z. B. JAVA) und Attribute (z. B. JavaScript) entfernt, da diese die Anonymität gefährden könnten. Attribute, die Adreßinformationen enthalten, werden zu absoluten Adressen erweitert und dann verschlüsselt.

5.4 Ver- und Entschlüsselung

Für die Ver- und Entschlüsselung von Adreßinformationen benutzen wir das RSA-Verschlüsselungssystem [RSA78] mit einem Schlüssel von 768 Bit. Als öffentlicher Schlüssel wird 65537 eingesetzt, um die Verschlüsselung zu beschleunigen.

Im Gegensatz zu Chaum verschlüsseln wir keine Zufallsbits, da der Inhalt unverschlüsselt übertragen wird. Vergleiche von verschlüsselten Adressen sind in unserem Fall erlaubt; schließlich kann ein Angreifer auch die Inhalte vergleichen, die er über JANUS beziehen kann.

6 Bekannte Probleme

6.1 Schutz vor Mißbrauch

Wir definieren Mißbrauch als die Benutzung von JANUS, die deutsches, europäisches oder internationales Recht verletzt, den Betrieb anderer Server oder Teile des Internets gefährdet oder gegen allgemeine Moralvorstellungen verstößt.

Leider kann das Interesse an mißbräuchlicher Nutzung der Server-Anonymität recht hoch sein: Beispielsweise hatte ein Händler, der mit illegalen Inhalten wie z. B. pornografischen Bildern handelt, bisher immer das Problem, daß seine Identität anhand der Internet-Adresse festgestellt werden konnte. Jetzt kann er unter Verwendung von Server-Anonymität mit seinen Inhalten handeln, ohne daß er deswegen verfolgt werden kann.

Um Mißbrauch zu verhindern,

- könnten wir die Server-Anonymität abschalten oder unsere geheimen Schlüssel veröffentlichen, um JANUS als wissenschaftliches Experiment ohne praktische Relevanz zu benutzen.

  Wir sind nicht bereit, das zu tun, da wir dann vielen legalen Benutzern den Dienst vorenthalten. Sollte es allerdings erforderlich werden, könnten wir durch häufige Schlüsselwechsel die Verbreitung verschlüsselter Adressen erschweren.

- könnten wir die ver- und entschlüsselte Adresse bekanntgeben, falls Mißbrauch auftrat. Somit können die Betreiber dieses Servers nicht mehr anonym arbeiten und hätten daher keinen Anreiz mehr, JANUS zu benutzen.

  Das Problem liegt jedoch darin, daß das Interesse an Listen mit Verweisen auf illegalen Inhalt recht hoch sein könnte und wir die Verbreitung solcher Adressen nicht vorantreiben wollen.

- werden wir auf Anfrage von Strafverfolgungsbehörden Adressen entschlüsseln oder sogar die Entschlüsselungsfunktion zur Verfügung stellen, ohne unsere geheimen Schlüssel bekannt zu geben.

- führen wir Ausschlußlisten, die wir nicht veröffentlichen. Bei jeder Anfrage vergleicht JANUS die Adresse mit allen Adressen in der Ausschlußliste und gibt bei Übereinstimmung eine entsprechende Meldung, nicht jedoch die gefragte Seite zurück.

Zusätzlich werden alle Zugriffe auf JANUS wie bei einem normalen Web-Server aufgezeichnet.

6.2 Regeln für anonymes Publizieren

Um zuverlässige Server-Anonymität zu erreichen, ist die Beachtung der folgenden Regeln von grundlegender Bedeutung:

- Adreßinformationen sollten nur dort eingesetzt werden, wo JANUS diese als solche erkennt (z. B. in den Href-Attributen der A-Tags), nicht jedoch im Text einer Seite.

- Die zu anonymisierenden Seiten sollten getestet werden, indem sie über JANUS geladen und dann anhand des HTML-Quelltextes überprüft werden.

- Der Zugriff von Suchmaschinen sollte unterbunden werden. Ein möglicher Angriff liegt darin, mit Hilfe von Stichwörtern einer Seite nach dieser Seite zu suchen, um bei Erfolg die Adresse des Servers bestimmen zu können.

- In der Zukunft sollten zwei vertrauenswürdige JANUS-Server benutzt werden.

- Es sollte kein illegaler Inhalt veröffentlicht werden, da die Adresse andernfalls auf Anforderung von Strafverfolgungsbehörden entschlüsselt wird.

7 Zusammenfassung und Ausblick

Nachdem Anonymität im Fall von Email bereits in mehreren Ansätzen analysiert und realisiert wurde, arbeiten zur Zeit einige Projekte am Thema Anonymität im World Wide Web. Die meisten davon konzentrieren sich auf die Client-Anonymität, und einige beziehen auch die Unbeobachtbarkeit der Kommunikationsbeziehung mit in ihre Überlegungen ein. In diesem Papier wird erstmalig ein neuer Ansatz zur Realisierung von Server-Anonymität vorgestellt und ein entsprechender Dienst im World Wide Web angeboten.

Da das Interesse, Server-Anonymität zu mißbrauchen, recht hoch sein kann, haben wir uns darauf beschränkt, nur die Adreßinformationen zu verschlüsseln und somit von dem Konzept

von Chaum abzuweichen. Wir sind daher in der Lage, Mißbrauch zu erkennen und zu verhindern.

Der JANUS-Prototyp basiert auf HTTP, HTTPS, FTP und Gopher als Transportprotokoll. Ein HTML-Parser ist in der Lage, Adreßinformationen aus dem Inhalt zu extrahieren, um diese mit einem RSA-Verschlüsselungssystem zu verschlüsseln.

Nachdem der JANUS-Prototyp im November 1997 fertiggestellt wurde, empfängt er heute durchschnittlich 3000 Anfragen am Tag. Wir haben bis heute keinerlei Hinweise auf Mißbrauch des Prototyps erhalten.

In Abhängigkeit von weiteren Erfahrungen planen wir zur Zeit, ein Netzwerk mehrerer JANUS-Server in verschiedenen Ländern aufzubauen. Diese werden Schritt für Schritt erweitert (Verschlüsselung des Inhaltes, Variation der Nachrichtenlänge, Verzögerung der Nachrichten, Erzeugung von Schein-Nachrichten, ...), um dem Konzept von Chaum näher zu kommen.

Im Gegensatz zum Chaum-Konzept möchten wir das wiederholte Abrufen von Informationen (z. B. aus einer Bookmark-Liste) ermöglichen, da dies den praktischen Umgang mit dem System wesentlich erleichtert. Technische Lösungen für diese Anforderung existieren, indem jeder JANUS einen Zwischenspeicher (Cache) betreibt, und wiederholte Anfragen daraus beantwortet. Rechtlich gesehen ist dieses Vorgehen jedoch problematisch, so daß wir hier ebenfalls noch an Lösungen arbeiten.

Diese Arbeit wurde am Forschungsinstitut für Telekommunikation (FTK) und am Lehrgebiet Kommunikationssysteme der FernUniversität Hagen unter Aufsicht von Prof. Dr.-Ing. F. Kaderali durchgeführt. Wir bedanken uns bei Prof. Kaderali, Prof. Pfitzmann und seinen Mitarbeitern und unseren Kollegen für viele interessante Diskussionen.

Literatur

[BGG+98] Bleichenbacher, Daniel; Gabber, Eran; Gibbons, Phillip B.; Matias, Yossi; Mayer, Alain: On Secure and Pseudonymous Client-Relationships with Multiple Servers / Bell Labs – Lucent Technologies. Murray Hill, NJ, Mai 1998

[BLFF96] Berners-Lee, T.; Fielding, R.; Frystyk, H.: Hypertext Transfer Protocol – HTTP/1.0. Mai 1996. – RFC 1945

[Cha81] Chaum, David L.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM 24 (1981), Februar, Nr. 2, S. 84-88

[GGMM97] Gabber, Eran; Gibbons, Phillip B.; Matias, Yossi; Mayer, Alain: How to Make Personalized Web Browsing Simple, Secure, and Anonymous. In: Hirschfeld, Rafael (Hrsg.): Proceedings of Financial Cryptography '97. Berlin: Springer, 1997 (LNCS 1318), S. 17-31

[PW87] Pfitzmann, Andreas; Waidner, Michael: Networks without User Observability. Computers & Security 6 (1987), S. 158-166

[Rag97] Raggett, Dave: HTML 3.2 Reference Specification / W3C. 1997. – W3C Recommendation

[RR97] Reiter, Michael K.; Rubin, Aviel D.: Crowds: Anonymity for Web Transactions / AT&T Labs. Murray Hill, NJ, August 1997. – DIMACS Technical Report 97-15

[RSA78] Rivest, R. L.; Shamir, A.; Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21 (1978), Februar, Nr. 2, S. 120-126

[RSG98] Reed, Michael G.; Syverson, Paul F.; Goldschlag, David M.: Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communications 16 (1998), Mai, Nr. 4, S. 482-494

[SRG97] Syverson, Paul F.; Reed, Michael G.; Goldschlag, David M.: Private Web Browsing. Journal of Computer Security Special Issue on Web Security 5 (1997), Nr. 3, S. 237-248