

BAMBERGER BEITRÄGE
ZUR WIRTSCHAFTSINFORMATIK UND ANGEWANDTEN INFORMATIK
ISSN 0937-3349

Nr. 75

**Extended Abstracts of the
Second Privacy Enhancing Technologies
Convention (PET-CON 2008.1)**

Karsten Loesing (Ed.)

Aachen, Germany, February 11, 2008

FAKULTÄT WIRTSCHAFTSINFORMATIK UND ANGEWANDTE INFORMATIK
OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG

Preface

PET-CON, the Privacy Enhancing Technologies Convention, is a forum for researchers, students, developers, and other interested people to discuss novel research, current development and techniques in the area of Privacy Enhancing Technologies. PET-CON was first conceived in June 2007 at the 7th International PET Symposium in Ottawa, Canada. The idea was to set up a bi-annual convention in or nearby Germany to be able to meet more often than only once a year at some major conference.

The First Privacy Enhancing Technologies Convention (PET-CON 2007) took place on August 16, 2007 in Frankfurt (on the Main), Germany. There were participants from four German universities.

The Second Privacy Enhancing Technologies Convention (PET-CON 2008.1) is held on February 11, 2008 in Aachen, Germany. This is the first time that we accept submissions, provide reviews, and publish a booklet of Extended Abstracts. Submitting a contribution is not mandatory for participating in PET-CON, and no submissions were rejected. All submissions were gratefully revised by three anonymous reviewers each.

We would like to thank all authors for submitting an Extended Abstract, all reviewers for conducting their work on really short notice, the Distributed and Mobile Systems Group at University of Bamberg for funding this booklet of Extended Abstracts, and the organizers at RWTH Aachen to make the actual convention possible.

February 2008

Karsten Loesing
Editor

Reviewers

Lothar Fritsch, Norwegian Computing Center, Norway
Dominik Herrmann, University of Regensburg, Germany
Karsten Loesing, University of Bamberg, Germany
Sebastian Pape, University of Kassel, Germany
Lexi Pimenidis, RWTH Aachen, Germany
Johannes Renner, RWTH Aachen, Germany
Jan Zibuschka, University of Frankfurt, Germany

Organizers

Andriy Panchenko, RWTH Aachen, Germany
Lexi Pimenidis, RWTH Aachen, Germany

Contents

1	Website-Fingerprinting mit dem multinomialen Naïve-Bayes-Klassifizierer <i>Dominik Herrmann</i>	1
2	Tor Hidden Services for Closed User Groups <i>Karsten Loesing</i>	9
3	Embedding Biometric Information into Anonymous Credentials <i>Sebastian Pape</i>	15
4	Introducing Measurable Path Selection Metrics to Anonymizing Overlay Networks <i>Johannes Renner</i>	22
5	A volume-based Accounting System for fixed-route Mix Cascade Systems <i>Rolf Wendolsky</i>	26
6	Ein einfaches Anonymisierungsverfahren basierend auf offenen Standards <i>Benedikt Westermann</i>	34

Website-Fingerprinting mit dem multinomialen Naïve-Bayes-Klassifizierer

Dominik Herrmann
Lehrstuhl Management der Informationssicherheit
Universität Regensburg, Deutschland
dh@exomail.to

Zusammenfassung

In diesem Arbeitspapier wird ein verbessertes Verfahren zur Identifizierung von Webseiten anhand des charakteristischen Datenverkehrs, der bei ihrem Abruf entsteht, vorgestellt und mit Testdaten evaluiert. Es basiert auf dem multinomialen Naïve-Bayes-Klassifizierer, der auf die normalisierte Häufigkeitsverteilung der IP-Paketgrößen angewendet wird. Das Verfahren ist bereits bei einer einzigen Trainingsinstanz mit einer Erkennungsrate von 89 % genauer als bislang vorgestellte Methoden. Darüber hinaus werden eine Reihe von Forschungsfragen formuliert, die zur Evaluierung der Praktikabilität von Website-Fingerprinting zu untersuchen sind.

1 Einleitung

Datenschutzfreundliche Übertragungsverfahren (z. B. Protokolle zur verschlüsselten Kommunikation in drahtlosen Netzen, VPNs sowie Anonymisierungssysteme wie JonDonym¹ und Tor² sollen die Inhalte der übermittelten Nachrichten sowie u. U. die Identitäten von Sender und/oder Empfänger vor Außenstehenden verbergen.

Beim Abruf einer Webseite über solche Systeme ist dieser Schutz jedoch möglicherweise schwächer als angenommen. Die zur Übermittlung der einzelnen HTTP-Anfragen und -Antworten ausgetauschten IP-Pakete ergeben ein charakteristisches Profil, das auch durch den Einsatz von Verschlüsselung nicht eliminiert wird. Anhand von einzelnen Netzwerk-*traces*, die zu *Fingerabdrücken* kombiniert werden, lässt sich dann die Identität (URL) einer Webseite ermitteln.

Dieses Arbeitspapier hat zwei Ziele: Zunächst soll ein verbessertes Verfahren zur Erstellung von Website-Fingerabdrücken vorgestellt werden, das die Häufigkeitsverteilung der beim Abruf einer Website beobachteten IP-Paketgrößen mit einem multinomialen Naïve-Bayes-Klassifizierer

¹<http://www.jondonym.de/>; vormals AN.ON bzw. JAP

²<http://tor.eff.org/>

analysiert. Da die Genauigkeit der Klassifizierung von Webseiten unter Idealbedingungen inzwischen sehr hoch ist, ist zukünftig neben der Untersuchung von Gegenmaßnahmen vor allem von Interesse, inwiefern sich Website-Fingerprinting *in der Praxis* durchführen lässt. Hierzu werden im zweiten Teil Forschungsfragen formuliert, deren Untersuchung zu einer besseren Einschätzung des Gefahrenpotentials, das von Website-Fingerprinting-Angriffen ausgeht, führen soll.

Abschnitt 2 gibt einen Überblick über verwandte Arbeiten. Abschnitt 3 präsentiert das unterstellte Angreifermodell, während Abschnitt 4 auf den Versuchsaufbau eingeht, der zur Erstellung von Website-Fingerabdrücken verwendet wurde. In Abschnitt 5 werden das verbesserte Verfahren zum Vergleich von Fingerabdrücken erläutert sowie erste Ergebnisse präsentiert. Die offenen Forschungsfragen werden in Abschnitt 6 beschrieben.

2 Verwandte Arbeiten

Bissias et al. [BLJL05] verwenden den Korrelationskoeffizienten, um Webseiten anhand ihres charakteristischen Datenverkehrs zu identifizieren. Die Autoren betrachten dabei die auftretenden IP-Paketgrößen sowie die zeitlichen Abstände zwischen den Paketen (*packet inter-arrival time*), wobei die Reihenfolge der Pakete berücksichtigt wird. Die Genauigkeit des Verfahrens wird durch den Abruf von 100 populären Seiten über einen OpenSSH-Tunnel getestet. Der dabei entstehende Datenverkehr wird mit *tcpdump* aufgezeichnet. Zur Erzeugung der Fingerabdrücke kombinieren die Autoren die Merkmale, die sie aus den *tcpdump-traces* extrahiert haben. Bei Verwendung von 24 *traces*, die innerhalb eines Tages aufgenommen wurden, werden nach einer Stunde nur 23 % der Seiten richtig identifiziert. Immerhin befinden sich etwa 60 % der Seiten jeweils unter den zehn ähnlichsten Profilen.

Liberatore et al. [LL06] stellen ein verbessertes Verfahren vor, das sie ebenfalls mit einem OpenSSH-Tunnel evaluieren. Zur Erstellung der Fingerabdrücke verwenden sie ausschließlich die beobachteten IP-Paketgrößen während der Übertragung einer Webseite. Die Fingerabdrücke werden mit dem *Jaccard-Koeffizienten*, einer Ähnlichkeitsmetrik für Mengen, die die Häufigkeiten vernachlässigt, sowie einem *Naïve-Bayes-Klassifizierer* miteinander verglichen. Die Genauigkeit der beiden Verfahren ist vergleichsweise hoch: Von 1000 zu identifizierenden Seiten werden bei Verwendung von Fingerabdrücken, die lediglich aus einem einzigen *trace* bestehen, mit dem Jaccard-Koeffizienten nach 24 Stunden noch 60 % der Seiten korrekt erkannt. Der Naïve-Bayes-Klassifizierer erzielt im direkten Vergleich lediglich eine Genauigkeit von 40 %. Erst bei Fingerabdrücken, die aus acht *traces* bestehen, erreicht der Naïve-Bayes-Klassifizierer eine zufrieden stellende Performance (ca. 75 % der Seiten werden korrekt erkannt). Beide Verfahren tolerieren die typische Änderung von Webseiten im Zeitverlauf. Selbst vier Wochen nach der Aufnahme der Fingerabdrücke sinken die Erkennungsraten nur um etwa 10 %.

Darüber hinaus zeigen die Autoren, dass die Effektivität beider Verfahren durch den Einsatz von Padding erheblich reduziert werden kann: Werden etwa alle IP-Pakete auf die MTU (1500 Byte) aufgefüllt, fallen die Erkennungsraten auf weniger als 10 %. Der Naïve-Bayes-Klassifizierer schneidet zwar erheblich besser ab als der Jaccard-Koeffizient, die niedrigen Erkennungsraten machen den Angriff jedoch unpraktikabel. Die Kosten des Paddings sind allerdings hoch: Das übertragene Datenvolumen nimmt dabei um mehr als 140 % zu.

3 Angreifermodell und Ziele des Angreifers

Für die nachfolgenden Betrachtungen wird von einem passiven lokalen Angreifer ausgegangen, der Zugriff auf das Netzwerk seines Opfers hat. Das Opfer verwendet zum Surfen ein datenschutzfreundliches Übertragungsverfahren (z. B. einen OpenSSH-Tunnel, ein VPN, Tor oder JonDonym). Der Angreifer will ermitteln, welche Webseiten das Opfer abgerufen hat bzw. ob eine ganz bestimmte Webseite abgerufen wurde. Es wird weiterhin unterstellt, dass die Pakete nicht entschlüsselt werden können. Der Angreifer ist lediglich in der Lage, den verschlüsselten Datenverkehr (gegebenenfalls auch über einen längeren Zeitraum) mitzulesen und auszuwerten.

Das Angreifermodell vieler datenschutzfreundlicher Techniken lässt einen solchen *lokalen Angreifer* durchaus zu. Nur ein stärkerer (verteilter) Angreifer, der z. B. den Datenverkehr vor und hinter einem Anonymisierer korreliert, kann bei diesen Systemen die Sender- bzw. Empfängeridentitäten ermitteln und verknüpfen. Durch Website-Fingerprinting kann diese Fähigkeit auch ein schwächerer (lokaler) Angreifer erlangen.

Je nach Zielsetzung werden unterschiedliche Anforderungen an ein Website-Fingerprinting-Verfahren gestellt. Zum einen kann ein Angreifer auf die Gewohnheiten und Interessen eines Benutzers schließen, obwohl dieser datenschutzfreundliche Techniken verwendet. Für eine solche Totalüberwachung muss der Angreifer Fingerabdrücke für eine möglichst große Anzahl von Webseiten aufzeichnen, die dann im abgehörten Datenverkehr des Opfers zu suchen sind. In Summe sind möglichst viele Webseiten zu identifizieren. Im Rahmen der Strafverfolgung kann jedoch auch eine andere Fragestellung auftreten: Es ist zu ermitteln, ob ein bestimmter Benutzer eine ganz bestimmte (möglicherweise inkriminierende) Webseite abgerufen hat. Im Vergleich zur Totalüberwachung ist die Anzahl der vorzuhaltenden Fingerabdrücke gering – diese wenigen Seiten sollen dann jedoch möglichst zuverlässig identifiziert werden.

4 Versuchsaufbau

Zur Analyse von Website-Fingerabdrücken wird folgender Versuchsaufbau verwendet: Auf einem Linux-Rechner wird mit Hilfe eines Skripts eine Instanz von Firefox automatisiert, so dass sie nacheinander eine Reihe von Webseiten abrufen. Wie bei bisherigen Untersuchungen ist zur Absicherung der Kommunikation im Browser ein OpenSSH-SOCKS-Proxy eingetragen.

Die Konfiguration des Browsers folgt den Beschreibungen in [BLJL05] und [LL06] (u. a. deaktiviertes Caching, keine automatischen Updates), wobei zusätzlich alle aktive Inhalte (Java, JavaScript, Flash, usw.) deaktiviert wurden. Die daraus resultierende Konfiguration ist mit der des JonDoFox-Browsers³, der für datenschutzfreundliches Surfen optimiert wurde, vergleichbar.

Bei jedem Abruf werden mit *tcpdump* die Header der übertragenen IP-Pakete protokolliert, um deren Paketgrößen zu ermitteln. Die Multimenge der auftretenden Paketgrößen wird im folgenden als *trace* bezeichnet. Abbildung 1 zeigt das Paketgrößen-Histogramm, das aus einem *trace* von `www.google.com` erstellt wurde.

³<https://www.jondos.de/en/jondofox>

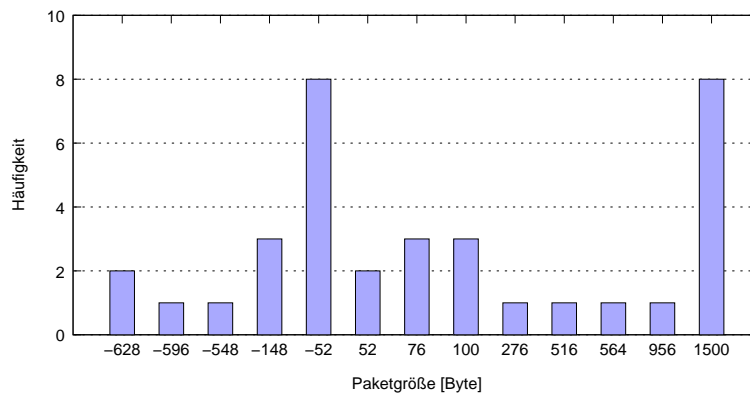


Abbildung 1: Paketgrößen-Histogramm für den Abruf von `www.google.com` (Pakete in Sende-richtung – vom Client zum Server – sind durch negative Paketgrößen gekennzeichnet.)

Zur Analyse der Genauigkeit des Klassifizierungsverfahrens wurden 775 URLs verwendet, die sich unter den populärsten Seiten befinden, die im Zeitraum von 12 Monaten über den vom Autor betriebenen Internet-Filter FilterSurf⁴ abgerufen wurden. Die URLs wurden manuell ausgewählt, um eine hohe Datenqualität sicherzustellen. Zur Analyse des Klassifizierungsverfahrens wurden die Webseiten im Zeitraum vom 09.01.2008 bis 19.01.2008 ohne Unterbrechung nacheinander heruntergeladen, wobei jede Webseite etwa zehnmal pro Tag abgerufen wurde.

5 Klassifizierung von *traces* mit Text-Mining-Techniken

Die Identifizierung von Webseiten anhand ihrer Fingerabdrücke erfolgt mit Hilfe von Weka⁵, einer Data-Mining-Software der University of Waikato. Zur Auswertung der Genauigkeit werden die beim Abruf erstellten *tcpdump-traces* in *Trainingsinstanzen*, mit denen der Klassifizierungsalgorithmus den Fingerabdruck einer Seite lernt, und *Testinstanzen*, deren Identität der Klassifizierer ermitteln soll, aufgeteilt. Die im Folgenden vorgestellten Ergebnisse wurden jeweils mit 10 Testinstanzen ermittelt. Für jeden Versuch wurden aus der Grundgesamtheit 25 Stichproben gezogen und der Mittelwert der Ergebnisse gebildet.

Während in [LL06] der *NaïveBayes*-Klassifizierer von Weka (mit *Kernel-Density-Estimation*) verwendet wird, setzt das in diesem Arbeitspapier vorgestellte Verfahren den *Naïve-Bayes-Multinomial*-Klassifizierer (vgl. [MRS07]) in Verbindung mit dem *StringToWordVector*-Filter von Weka ein – eine Kombination, die klassischerweise im Text-Mining bei der Klassifizierung von Text-Dokumenten verwendet wird. Ein *trace* wird dabei als String repräsentiert, der aus den aufgetretenen Paketgrößen (durch Leerzeichen getrennt) besteht (z. B. „-628 956 -52 1500 1500 -52 -52 1500 ...“). Der *StringToWordVector*-Filter ermittelt aus einem solchen Dokument die Termhäufigkeiten, also die Auftretenshäufigkeiten der einzelnen Paketgrößen. Für jedes Dokument (also für jeden *trace*) ergibt sich dann ein Häufigkeitsvektor, dessen Elemente den jeweiligen Termhäufigkeiten (oder 0, falls der Term nicht vorkommt) entsprechen.

⁴<http://www.filtersurf.de/>

⁵<http://www.cs.waikato.ac.nz/ml/weka/>

Der multinomiale Naïve-Bayes-Klassifizierer schätzt die Wahrscheinlichkeit, dass ein Dokument (*trace*) d , zu einer Klasse (*Webseite*) c gehört, nach folgender Formel:

$$\hat{P}(c|d) = \frac{\hat{P}(c)\hat{P}(d|c)}{\hat{P}(d)} \propto C_{\text{multi}} \cdot \prod_{t \in V} \left(\frac{f_{t,c}}{\sum_{t' \in V} f_{t',c}} \right)^{f_{t,d}} \quad (1)$$

Zur Klassifizierung eines Dokuments werden für alle Klassen $c \in C$ die Wahrscheinlichkeiten $\hat{P}(c|d)$ geschätzt. Das Dokument wird dann der Klasse mit der größten Wahrscheinlichkeit zugeordnet. Bei Bedarf lässt sich auch die Wahrscheinlichkeit dieser Zuordnung bestimmen.

Zur Klassifizierung ist die exakte Wahrscheinlichkeit nicht von Interesse. Daher lässt sich der mittlere Term in Formel (1) wie dort abgebildet vereinfachen. Sind alle Klassen (Webseiten) in der Grundgesamtheit gleichverteilt, kann die (dann einheitliche) Auftretenswahrscheinlichkeit der Klassen $\hat{P}(c)$ vernachlässigt werden. Die Auftretenswahrscheinlichkeit des Dokuments $\hat{P}(d)$ ist zur Ermittlung der wahrscheinlichsten Klasse ebenfalls bedeutungslos (da konstant) und muss nicht berücksichtigt werden. Der Term auf der rechten Seite der Gleichung ist dann immer noch proportional zur tatsächlichen Wahrscheinlichkeit; die Rangfolge der Wahrscheinlichkeiten $\hat{P}(c|d)$ bleibt dadurch erhalten.

In Formel (1) ist V das Vokabular aller vorkommenden Terme, $f_{t',c}$ die Auftretenshäufigkeit von Term t' in allen Dokumenten, die zur Klasse c gehören und $f_{t,d}$ die Auftretenshäufigkeit von Term t im Dokument d . Ein Dokument wird dabei als „bag of words“ aufgefasst, aus der die einzelnen Terme gezogen werden. Der zugehörige Multinomialkoeffizient $C_{\text{multi}} = \frac{L_d!}{f_{t_1,d}! \dots f_{t_n,d}!}$ ist für ein gegebenes Dokument d mit Länge L_d eine Konstante – er muss zur Ermittlung der wahrscheinlichsten Klasse gar nicht berechnet werden.

Dem „bag of words“-Modell liegen zwei naive Annahmen zugrunde, die bei Textdokumenten üblicherweise verletzt sind: (1) Die Reihenfolge der Terme (bzw. Paketgrößen) spielt keine Rolle und (2) das Vorkommen der einzelnen Terme ist voneinander unabhängig. Es ist davon auszugehen, dass diese Annahmen auch bei *tcpdump-traces* verletzt werden – der multinomiale Naïve-Bayes-Klassifizierer erzielt jedoch auch unter solchen Umständen in der Regel gute Erkennungsraten.

Die Interpretation eines Fingerabdrucks als Text-Dokument ermöglicht die Verwendung von einschlägigen Optimierungsmöglichkeiten aus dem Text-Mining. Durch geeignete Transformation der Häufigkeitsvektoren \mathbf{f} lassen sich die Erkennungsraten erheblich verbessern. Abbildung 2 zeigt die Klassifizierungsergebnisse für verschiedene Transformationen mit und ohne Normalisierung für den Fall, dass lediglich eine einzige Trainingsinstanz verwendet wird und zwischen Training und Test mindestens 6 Tage liegen.

Im Versuch zeigte sich, dass sich durch Verwendung der TF-Transformation $f_i^* = \log(1 + f_i)$ [MRS07] und die Normalisierung der Häufigkeitsvektoren auf die durchschnittliche euklidische Länge $f_i^{\text{norm}} = \frac{f_i^*}{\|(f_1^*, \dots, f_n^*)^T\|}$ [MRS07] die besten Ergebnisse erzielen lassen. Die IDF-Transformation, die beim Text-Mining denjenigen Termen ein höheres Gewicht zuordnet, die in einem Dokument besonders häufig vorkommen, in den anderen Dokumenten hingegen sehr selten sind, hat sich hingegen als kontraproduktiv erwiesen.

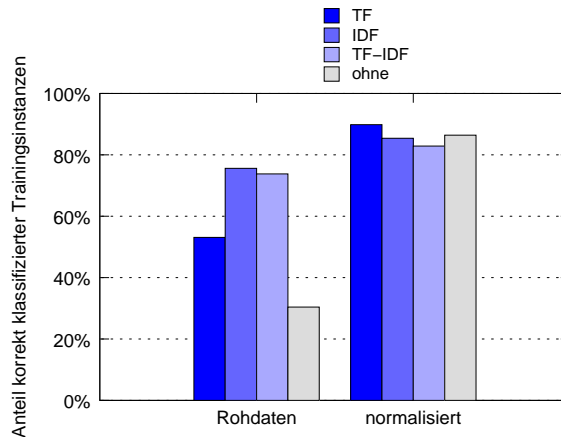


Abbildung 2: Einfluss verschiedener Transformationen auf die Genauigkeit der Klassifizierung bei 6 Tagen Zeitdifferenz zwischen Training und Test (bei einer einzigen Trainingsinstanz).

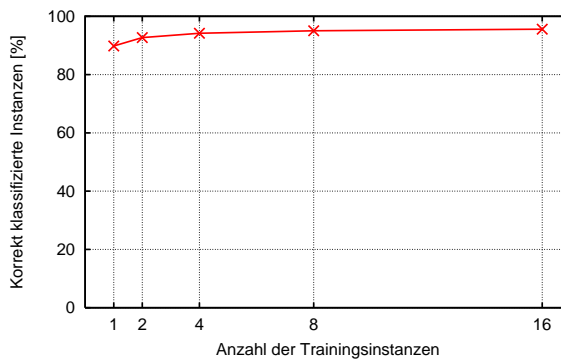


Abbildung 3: Einfluss der Anzahl der Trainingsinstanzen auf die Genauigkeit des Klassifizierers

Der multinomiale Naïve-Bayes-Klassifizierer erreicht bei einer Trainingsinstanz im besten Fall eine Erkennungsrate von 89,89%. Erhöht man die Anzahl der Trainingsinstanzen, nehmen die Erkennungsraten weiter zu. Abbildung 3 zeigt die Ergebnisse für verschiedene Varianten dieses Experiments. Bei vier Trainingsinstanzen werden bereits 94,18% der Testinstanzen korrekt identifiziert; bei mehr als vier Trainingsinstanzen sind die Zuwächse hingegen sehr gering.

Der multinomiale Naïve-Bayes-Klassifizierer übertrifft damit die Genauigkeit der bisher vorgestellten Verfahren. Zum Vergleich: in [LL06] wurden bei einer Grundgesamtheit von 1000 Seiten nach einer Woche nur ca. 70% der Seiten korrekt identifiziert. Die Aussagekraft eines solchen Vergleichs wird durch die unterschiedlichen Webseiten zwar leicht eingeschränkt, die Unterschiede sind jedoch angesichts der vergleichbaren Untersuchungsbedingungen nicht zu vernachlässigen.

Da sich das gezeigte Verfahren genauso wie seine Vorgänger auf die charakteristische Häufigkeitsverteilung der IP-Paketgrößen stützt, ist davon auszugehen, dass die Erkennungsleistung beim Einsatz von Padding deutlich sinkt. Eine erste Implementierung wurde unter dem Namen *Traffic Flow Confidentiality* für IPsec gerade erst von Kiraly et al. [KTB⁺07] vorgestellt.

6 Weiterer Untersuchungsbedarf

Im Folgenden werden verschiedene Forschungsfragen formuliert, deren Untersuchung einen Beitrag zur Evaluation von Website-Fingerprinting in der Praxis liefern könnte.

Skalierbarkeit Website-Fingerprinting kann zum Erstellen von Benutzerprofilen verwendet werden, wenn Fingerabdrücke für eine große Anzahl von Webseiten erstellt werden. Es ist zu klären, inwiefern das vorgestellte Klassifizierungsverfahren im Hinblick auf die Anzahl der Instanzen skaliert.

Einsatzmöglichkeiten Neben der Erstellung von Benutzerprofilen sind weitere Einsatzmöglichkeiten für Website-Fingerprinting denkbar. Zur strukturierten und zielgerichteten Analyse ist es erforderlich, potentielle Anwendungen zu kennen und explizit zu beschreiben.

Vergleich verschiedener datenschutzfreundlicher Techniken Die meisten Analysen widmen sich SSH-Tunneln, die praktisch kein Padding implementieren. Ein Vergleich mit anderen datenschutzfreundlichen Übertragungstechniken (z. B. SSL- und IPsec-VPNs) steht noch aus. Unter Umständen bieten die heute bereits verfügbaren Anonymisierungssysteme bereits genügend Schutz gegen Website-Fingerprinting.

Untersuchung effizienter Gegenmaßnahmen Bislang konzentriert sich die Entwicklung von Gegenmaßnahmen auf verschiedene Padding-Schemata. Wirkungsvolles Padding erhöht jedoch das übertragene Datenvolumen erheblich. Effizientere Gegenmaßnahmen (z. B. ein Burst-Proxy, der selbständig die in HTML-Seiten eingebetteten Objekte herunterlädt und in einem zusammenhängenden Datenstrom zum Client sendet) wurden zwar vorgeschlagen, jedoch noch nicht implementiert und auf ihre Wirksamkeit hin untersucht.

Gezielte Erleichterung von Website-Fingerprinting Es ist zu untersuchen, inwiefern der Betreiber einer Webseite bzw. der ISP deren Identifizierung mutwillig erleichtern kann. Solchermaßen modifizierter Traffic könnte mit Hilfe von Website-Fingerprinting auf dem Weg durch das Netzwerk verfolgt werden, um einem Nutzer den Besuch der manipulierten Webseite nachzuweisen. Darüber hinaus ist zu klären, ob es wirksame Gegenmaßnahmen zur Unterbindung dieses aktiven Angriffs gibt.

Überbrückung von Anonymisierungssystemen Möglicherweise lassen sich Anonymisierer wie Tor und JonDonym, die den Datenverkehr über mehrere Stationen weiterleiten, durch Website-Fingerprinting vor der ersten und nach der letzten Station überbrücken.

Erkennung von Website-Abrufen in getunneltem Traffic In den bisherigen Veröffentlichungen wird unterstellt, dass Beginn und Ende eines Seitenabrufs im verschlüsselten Datenverkehr wegen der Denkpausen des Benutzers leicht zu ermitteln sind. Aktuelle Studien [CCW⁺07, KAA06] deuten jedoch darauf hin, dass diese Annahme unzutreffend ist. Es ist demnach völlig ungewiss, ob Website-Fingerprinting in der Praxis (etwa wenn der Tunnel parallel auch von anderen Diensten genutzt wird) überhaupt durchführbar ist.

Robustheit Unterschiedliche Browser, Plugins (z. B. Ad-Blocker) und Internetverbindungen beeinflussen Art und Größe der übermittelten Pakete. Es wurde noch nicht untersucht, wie robust Website-Fingerabdrücke gegen solche äußeren Einflüsse sind.

Website-Fingerprinting beim Einsatz von Caching In bisherigen Studien war der Cache im Browser stets deaktiviert, um sicherzustellen, dass bei jedem Abruf einer Webseite alle eingebetteten Elemente übertragen werden. Es ist zu erwarten, dass die Erkennungsraten durch Caching erheblich sinken. Eine Analyse der Effektivität von Website-Fingerprinting beim Einsatz von Caching erlaubt Rückschlüsse auf die Einsetzbarkeit in der Praxis.

Bedeutung der False-Positive-Rate Bislang lag der Fokus bei der Analyse von Website-Fingerprinting auf Basis von IP-Paketgrößen auf den *True Positives* (Anzahl der korrekt identifizierten Webseiten). In [SSW⁺02] wird jedoch auf die hohe Relevanz der Minimierung der *False Positives* (fälschlich identifizierte Webseiten) hingewiesen. Eine Analyse des Klassifizierungsverhaltens für Webseiten, die nicht antrainiert wurden, steht noch aus.

Literatur

- [BLJL05] Bissias, George, Marc Liberatore, David Jensen, and Brian Neil Levine: *Privacy Vulnerabilities in Encrypted HTTP Streams*. In *Proceedings of the 5th Privacy Enhancing Technologies Workshop (PET 2005)*, pages 1–11, May 2005. <http://prisms.cs.umass.edu/brian/pubs/bissias.liberatore.pet.2005.pdf>.
- [CCW⁺07] Coull, S.E., M.P. Collins, C.V. Wright, F. Monroe, and M.K. Reiter: *On Web Browsing Privacy in Anonymized NetFlows*. In *Proceedings of the 16th USENIX Security Symposium*, Boston, MA, August 2007.
- [KAA06] Koukis, D., Spyros Antonatos, and Kostas G. Anagnostakis: *On the Privacy Risks of Publishing Anonymized IP Network Traces*. In Leitold, Herbert and Evangelos P. Markatos (editors): *Communications and Multimedia Security*, volume 4237 of *Lecture Notes in Computer Science*, pages 22–32. Springer, 2006, ISBN 3-540-47820-5.
- [KTB⁺07] Kiraly, Csaba, Simone Teofili, Giuseppe Bianchi, Renate Lo Cigno, Matteo Nardelli, and Emanuele Delzeri: *Traffic Flow Confidentiality in IPsec: Protocol and Implementation*. In *Preproceedings Third IFIP/FIDIS Summer School “The Future of Identity in the Information Society”*, August 2007.
- [LL06] Liberatore, Marc and Brian Neil Levine: *Inferring the source of encrypted HTTP connections*. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 255–263, New York, NY, USA, 2006. ACM Press, ISBN 1-59593-518-5.
- [MRS07] Manning, C. D., P. Raghavan, and H. Schütze: *Introduction to Information Retrieval (preliminary draft printed on November 17, 2007)*. Cambridge University Press, 2007. <http://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf>.
- [SSW⁺02] Sun, Qixiang, Daniel R. Simon, Yi Min Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu: *Statistical Identification of Encrypted Web Browsing Traffic*. In *SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy*, page 19, Washington, DC, USA, 2002. IEEE Computer Society, ISBN 0-7695-1543-6.

Tor Hidden Services for Closed User Groups

Karsten Loesing
Distributed and Mobile Systems Group
University of Bamberg, Germany
karsten.loesing@uni-bamberg.de

Abstract

Tor hidden services were designed to provide sender and responder anonymity for publicly accessible services. In this paper we motivate their use for closed user groups which implies revealing existence and permitting access only to authorized clients. We also sketch necessary changes to the existing hidden service protocol.

1 Introduction

Tor hidden services [DMS04] constitute a means of providing TCP-based services to clients without revealing the server's location in terms of IP address. Besides, hidden services promise to resist censorship and DDos attacks. This is referred to as *responder anonymity* and is harder to achieve than sender anonymity which protects the identity of a client using a service. There are often good reasons for people who provide potentially controversial services or content to others to hide their identity. Otherwise these people could be faced with personal consequences, ranging from job-related disadvantages up to prosecution and personal harm.

There has been a number of previous work on responder anonymity: The first design of Onion Routing [GRS96] introduced the concept of (possibly long-lived) *reply onions* which can be used together with an *anonymity server* to mate two virtual circuits and provide mutual anonymity. Goldberg [Gol00] described a design for privacy protection for servers that is based on the similar concept of a *rendezvous server* and in which service descriptions containing the rendezvous server's address are published in a Gnutella network. Tor hidden services [DMS04] evolved Goldberg's design to Tor's location-hidden services by including a *rendezvous point*, an *introduction point* for better DoS protection and an efficient key-value lookup system for hidden service descriptors on central *directory servers*. Øverlier and Syverson [ØS06] proposed an extension of Tor's hidden service protocol including so-called *contact information tickets* which allow for incorporating client authorization to a hidden service.

Tor hidden services were primarily designed to provide content on rather highly available servers to the anonymous public. An alternative way of using hidden services that we want to highlight

in this paper is to provide a service to a limited set of clients. Though being connected to the public Tor network, the intention is to hide all signs of existence of a hidden service from unauthorized clients and to perform access control. The service should not leak any information to the network of Tor relays and directories that could be associated with the service's identity. Further, it should be easy to remove authorization from clients, so that the service is again unaccessible for and its existence hidden from formerly authorized clients.

One possible application could be a *web or file service* that should only be accessible by a limited set of clients, e.g. a distributed working group. Especially when clients are semi-trusted, it makes sense to keep the service's location hidden and hide existence of the service again after a client's authorization is removed. Presence-aware communication media like *instant messaging* are a second and rather different application domain. Service providers offer a presence and messaging service to their buddies, but usually want to hide their presence from the public and avoid being annoyed by unsolicited messages from strangers. Particularly when it comes to communication, it might be important to gently remove authorization by hiding the existence of a service, rather than explicitly letting a client know about the removal. In [LDG⁺06, LRWW07] we proposed such a presence-aware instant messaging system. A third example is *running a service at home*, e.g. a video recording service, and accessing it while being under way. Such a service should be specifically protected from attacks, which is why opening a port to the public Internet should be avoided. Further, service availability could reveal the service provider's personal habits, because its uptime might correlate with the user's daily routine.

2 Problem statement

In order to comprehend our motivation it is necessary to understand the basic Tor hidden service protocol (cf. Figure 1): A hidden service initiates the protocol by requesting some Tor relays to act as his introduction points. Therefore he sends a request to them including his *permanent key* in step 1. Next, the hidden service creates a hidden service descriptor including his public permanent key, a *timestamp*, a list of introduction points, and a signature of these data, created with his private permanent key. The hidden service publishes this hidden service descriptor to the directory servers in step 2. Then, the hidden service tells his *onion address*, a hash of his permanent key, to his clients out of band in step 3. Whenever a client wants to access the hidden service, she looks up the current hidden service descriptor from a directory server by requesting an entry for the service's onion address in step 4. Assuming that she received a correct descriptor, she picks a random relay and requests it to act as rendezvous point on her behalf by sending it a single-use *rendezvous cookie* in step 5. Next, the client picks one introduction point from the hidden service descriptor and sends to it an introduction request in step 6. The introduction request consists of the unencrypted onion address and an encapsulated message that is encrypted using the service's public permanent key and that contains contact information of her rendezvous point, the rendezvous cookie, and the first half of a Diffie-Hellman handshake. If the introduction point recognizes the onion address, it forwards the rest of the message to its hidden service in step 7. If the hidden service wants to answer the request, he contacts the client's rendezvous point and sends it the rendezvous cookie and the second half of the Diffie-Hellman handshake in step 8. The rendezvous point forwards the request to the

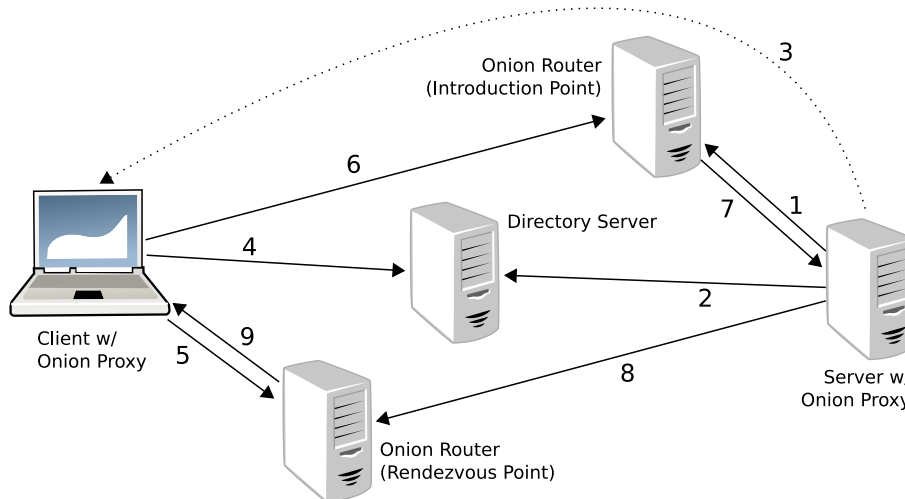


Figure 1: Tor hidden service protocol overview

client in step 9, finally establishing a connection between client and hidden service.

There are a number of *security problems* in this protocol with regard to hiding the hidden service identity and protecting the location of introduction points: In step 1, the introduction point gets to know the permanent key that clearly identifies the hidden service. After being an introduction point for a hidden service for a single time, a node can reliably track presence of the hidden service by periodically downloading his hidden service descriptor, learn about current introduction points, and even attempt to access the hidden service (unless it is secured by the transported protocol). Next, a directory server learns about a hidden service's permanent key in step 2 whenever the hidden service publishes a hidden service descriptor. This allows a directory server to keep track of a hidden service's presence, simply by passively keeping logs. Further, a directory server is told the current list of introduction points, enabling it to attack the introduction points or attempt to access the hidden service as any client could do. Although this might appear subtle, a directory server is also able to generate access patterns of anonymous clients to a certain hidden service by observing requests for hidden service descriptors. Finally, every client that learns about the onion address in step 3 will be able to request a current hidden service descriptor and attempt to access the hidden service from that time on. It is impossible for the hidden service to restrict access or hide existence from anyone who has once learned about his onion address.

When trying to achieve the postulated security properties, i.e. revealing existence and permitting access only to currently authorized clients, one might first conceive solutions that do not change the basic Tor hidden service protocol. For example a service provider could create a separate hidden service for each client to be able to selectively deny service. Unfortunately, this would lead to a linearly growing number of introduction circuits and hidden service descriptors and is thereby simply not efficient. Another solution could be to store the onion address of a periodically changing hidden service in an external place for each authorized client. However, such an approach would be equally complex as ours, but would require external resources and software programs. Our approach combines the ideas behind these simple solutions (plus some more), but aims to incorporate them in the core hidden service protocol.

3 Proposed changes

In the following we sketch a solution that restricts access to authorized clients and hides existence from all other entities in the network. Our proposal requires a couple of changes to the existing hidden service protocol, essentially consisting of four parts: First, an introduction point does not learn about the identity of a hidden service. Second, hidden service descriptors are stored on a subset of a large set of directory servers instead of a small, fixed number of them in order to make tracking of hidden services very hard. Third, the hidden service descriptor format shall reveal as few useful information as possible to the storing directory server. And fourth, every client authenticates herself to the hidden service so that the hidden service can remove any client's authorization and keep the service unchanged for the remaining clients.

Introduction point establishment As stated above, an introduction point can easily identify a hidden service from the permanent key that it receives upon establishment in step 1 of the hidden service protocol. However, an introduction point is not required to learn *the* identity of a hidden service, but only *an arbitrary* identity created by the hidden service to perform its task: match incoming client requests with a previously registered hidden service. Therefore, the client needs to learn about that identity before contacting the introduction point and be assured that it belongs to the hidden service. Further, the introduction point should make sure that the entity claiming to have a given identity really owns it and is not just replaying an old message from the real owner containing such a claim. The identity does not even need to persist for longer than a single introduction point establishment in order to prevent linkage between establishments. We propose to create a new introduction key for every introduction point establishment, replacing the permanent key for introduction, and include it in the hidden service descriptor.

Storage of hidden service descriptors The security problems also comprise the fact that directory servers know about all hidden service publications and (at least a part of) requests. A directory server could use these information to generate logs of presence and access patterns of a given hidden service. Although we propose a change to the hidden service descriptor format below to prevent revealing identifying information of a hidden service, we want to make tracking of a certain hidden service as hard as possible. Therefore, hidden service descriptors are stored on a small, periodically changing subset of a large number of *hidden service directories* which are specifically configured relays organized as a distributed hash table. Each hidden service directory is responsible for storing a small subset of hidden service descriptors for a limited time. Further objectives of distributing the directory of hidden service descriptors are improved scalability and robustness against node failures. Especially scalability is an issue in our design, because the overall number of hidden service descriptors is linear to the number of authorized clients, which cannot be handled by central directory servers any more.

Hidden service descriptor format The remaining security problems of the discussion above are related to the contents of hidden service descriptors, including their identifiers: directory servers/hidden service directories get to know the list of introduction points, and clients

will always be able to request a hidden service descriptor of a previously known hidden service.

The new descriptor ID is derived from an asymmetric *client key* instead of the hidden service's permanent key, a symmetric *descriptor cookie*, and a *time period*. It is calculated using the formula: $H(H(\text{client key}), H(\text{descriptor cookie}, \text{time period}))$. The client key is only used for a single client or a group of to-be-indistinguishable clients. The hidden service publishes a distinct hidden service descriptor for each client, which is why we require the directory storage to be truly scalable. A hidden service distributes the hash value of the client key and the descriptor cookie in his message to a client outside of Tor, making the original onion address dispensable. The reasons for using a client key instead of the permanent key is that a previously authorized client that acts as or cooperates with a hidden service directory cannot learn the presence of a hidden service. The new descriptor ID also prevents a hidden service directory from calculating the next descriptor ID from a given descriptor ID or hidden service descriptor to prevent tracking of hidden services: Due to the one-way property of the hash function, a hidden service directory that does not know the descriptor cookie is unable to calculate the descriptor ID for another time period. Next, the descriptor ID changes periodically due to inclusion of the time period, so that a hidden service descriptor is stored on changing directories over time.

In the new hidden service descriptor content the hidden service's permanent key is replaced with the client key, the result of the inner hash function for calculating the descriptor ID is included, the introduction points contain the introduction keys as described above, and the introduction points are encrypted using the descriptor cookie. First of all, clients are still able to validate the hidden service as origin of an hidden service descriptor by verifying the signature of a hidden service descriptor with the contained client key. However, it is a little bit harder to enable a hidden service directory to authenticate the creator of the hidden service descriptor as the entity that the client expects when requesting a specific descriptor ID. If the hidden service directory would not perform any validation, anyone could store arbitrary data for a given descriptor ID. Validation is performed in two steps: First, the hidden service directory verifies that the signature of the hidden service directory was created with the included client key. Second, it verifies that the descriptor ID can be recreated from the client key and the result of the inner hash function. If both verifications succeed, the hidden service directory can be sure that the hidden service descriptor was created by the entity owning the private client key *and* that the hidden service descriptor is permitted to be stored under the stated descriptor ID. An adversary trying to store a different descriptor content for a given descriptor ID would necessarily have to use another client key in order to create a valid signature. But due to the one-way property of hash functions, this adversary would not be able to determine a matching result of the inner hash function of the descriptor ID. Finally, only clients are able to read the encrypted introduction information using their descriptor cookie for decryption.

Authentication to hidden service The last change to the hidden service protocol relates to the ability of a hidden service to selectively remove authorization of a client. If clients would not authenticate themselves to a hidden service, the hidden service could never be able to attribute service misuse to one of his clients. Client authentication might sound contradictory to the goals of an anonymity system; but as soon as service authorization is granted or denied based on shared knowledge of a secret, a client knowing that secret is anyhow not anonymous any more. A hidden service could easily create a separate service for each client, thereby creating an

anonymity set of 1, and find out who is contacting him. This is why we provide the possibility to efficiently remove a client by including the descriptor cookie in an introduction request that is sent to the hidden service.

4 Conclusion

Recapitulating, we motivated the application of Tor hidden services for closed user groups. This requires to hide the service's existence from untrusted entities, including the formerly trusted clients, and to perform access control. We pointed out at which points the existing hidden service protocol conflicts with these requirements and sketched a solution that changes the current hidden service protocol. Some parts of these changes are already specified and implemented in Tor [Loe07]. The implementation of the remaining parts is subject to future work.

References

- [DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.
- [Gol00] Ian Goldberg. *A Pseudonymous Communications Infrastructure for the Internet*. PhD thesis, UC Berkeley, December 2000.
- [GRS96] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150, Cambridge, UK, May 1996. Springer.
- [LDG⁺06] Karsten Loesing, Markus Dorsch, Martin Grote, Knut Hildebrandt, Maximilian Röglinger, Matthias Sehr, Christian Wilms, and Guido Wirtz. Privacy-aware Presence Management in Instant Messaging Systems. In *20th IEEE International Parallel and Distributed Processing Symposium*, April 2006.
- [Loe07] Karsten Loesing. *Distributed Storage for Tor Hidden Service Descriptors*, May 2007. <https://tor-svn.freehaven.net/svn/tor/trunk/doc/spec/proposals/114-distributed-storage.txt>.
- [LRWW07] Karsten Loesing, Maximilian Röglinger, Christian Wilms, and Guido Wirtz. Implementation of an Instant Messaging System with Focus on Protection of User Presence. In *Proceedings of the Second International Conference on Communication System Software and Middleware*. IEEE CS Press, January 2007.
- [ØS06] Lasse Øverlier and Paul Syverson. Valet services: Improving hidden servers with a personal touch. In George Danezis and Philippe Golle, editors, *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, pages 223–244, Cambridge, UK, June 2006. Springer.

Embedding Biometric Information into Anonymous Credentials

Sebastian Pape

Databases and Interactive Systems Research Group

University of Kassel, Germany

pape@db.informatik.uni-kassel.de

Abstract

Anonymous credentials allow people to authenticate to an organisation without being identified. While there are some approaches trying to ensure non-transferability we want to improve the methods relying on biometric authentication. In this working paper we present a first step of embedding biometric attributes into anonymous credentials. There are only slight changes to existing approaches, but the main purpose of this working paper is to establish a basis, which can be improved in the future to solve some of the problems addressed in the “future work section”.

1 Introduction and Related Work

Anonymous credentials [CE87] introduced by CHAUM [Cha85] usually consist of cryptographic tokens which allow the user (herein after referred to as prover) to prove a statement or relationship with an organisation to another person or organisation (herein after referred to as verifier) without being identified. While some anonymous credential systems are related to the concept of untraceable or anonymous payments [Cha83] and hence it should be possible to transfer them easily to another person there are some situations where credentials should not be transferable. E.g. if the prover wants to show the possession of a valid driving licence the verifier probably does not want to see a transferred driving licence which would rather prove the statement “I know someone who has a valid driving licence”. There are already approaches to prevent the transfer of credentials where sharing a credential implies also sharing a valuable secret outside a system [DLN97, GPR98, LRSW00] or even all of the prover’s secrets inside the system [CL01]. Nevertheless that protection naturally will not prevent all users from sharing credentials. Be it they share their credentials incautious, be it they really trust someone else. In addition those valuable secrets raise the system’s value. So users have to take care of thieves and have to immanently trust the system’s architecture.

Another possibility to make sure the credentials are only used by the person the credential was created for, is to make use of the person’s biometric information. Using biometrics however

usually causes privacy concerns, especially since – in contrast to passwords or tokens – you cannot change biometric attributes. Therefore extraordinary care has to be taken to protect the user’s data. It can be easily seen that allowing the verifier to check the prover’s biometric attributes conflicts with the prover’s wish of anonymity. In 1998 BLEUMER [Ble98] combined anonymous credentials with biometric authentication making use of a variant of the *wallet-with-observer-architecture* introduced by CHAUM and PEDERSEN [CP93]. In the wallet-with-observer-architecture there exists a user trusted device (wallet) which runs a local process (observer). The credential issuing organization (herein after referred to as authority) trusts that the observer only performs legitimate operations. IMPAGLIAZZO and MINER MORE [IM03] transferred that design to a personal digital assistant (PDA) with a tamper-resistant smartcard. The smartcard (observer) is issued and trusted by the authority and its tamper-resistance makes sure that the user cannot read and tamper with its content. In contrast the PDA (warden) protects the users interests and makes sure the smartcard does not diverge from the specified protocol. Both approaches have in common that biometric authentication is not part of the underlying credential system, but instead prerequisite for the credential protocol to start.

The outline of this paper is as follows. In the next section we discuss which biometric attributes could fit our purpose. In section 3 we describe a sketch of our approach. Section 4 roughly compares our contribution to [IM03] and finally we present our future work and some open questions in the last section.

2 Finding Applicable Biometric Attributes

When embedding biometric information into anonymous credentials care has to be taken that the verifier is not able to identify the prover. This can be done by using biometric attributes which are not suitable to identify the user. Let us assume we had ten biometric attributes, each dividing the users independent of other attributes and randomly into five subgroups of equal size. E.g. haircolor (brown/black, blond, red, none, others) or iris color (blue, brown, green, grey, hazel). In total that would split the users into almost ten million subgroups. If we only check two attributes when verifying the user’s credential, the user cannot be identified if there are enough users participating in the system. On the other hand there is a good chance of catching users if they are using another person’s credentials. On average two attributes of users will be the same. Since the user does not know in advance which attribute will be checked, the probability of a successful abuse will be roughly four percent ($0.2 * 0.2$). Of course it may be increased a bit if the attributes are well known. But depending on the necessary level of security and the estimated penalty this could be enough for some situations (e.g. personalized bus or train tickets). Unfortunately taking a closer look at biometrics reveals that it could be a problem of finding enough suitable attributes. The abovementioned hair and iris color differ hardly outside europe [SF04]; brown is very predominant and, in many populations, it is (with few exceptions) the only iris color present [FDH03]. Other attributes like skin tone or seize have more fluent bounds, may vary too much (sun, aging) or they are closer to normal distribution than to equal distribution.

Therefore the only approach seems to use a biometric attribute which is suitable to identify

the user and to cut down the information seen by the verifier. That means, that neither the verifier nor the user must have the biometric device under their control. The first puts the user's privacy at risk, while the latter compromises the non-transferability of the tokens.

3 Sketch of our Contribution

Analogous to [IM03] we prefer a setup where each user has a PDA and a smartcard handed out by the credential issuing authority. Following the previous section renders it obvious that the biometric device has to be connected straightly to the smartcard. The only devices which fulfill our needs today are fingerprintreaders. Either they are embedded into the smartcard [Bio, fid08] or they work the way cash-card-terminals act when asking for the users personal identification number (PIN). In this case the reader's input is directly sent to the card, the smartcard decides about acception or rejection (called *match-on-card* [Nor04, NH04]) and no attached computer is able to eavesdrop on it.

Note that relying on fingerprints is only a compromise solution. On the one hand some people do not have suitable fingerprints and forging fingerprints has been done with acceptable effort. On the other hand today fingerprint readers are the only devices which could be embedded into smartcards. Following the parameters and comparison of [Jai04] the most desirable attributes for us would be to have a low *Circumvention* (the ease of use of substitutes is hard) and high *Universality* (as many people as possible have those attributes), *Uniqueness* (people can be well seperated) and *Permanence* (the attributes resist aging well). That would lead us to the use of DNA-recognition which can't be done on a smartcard. Since we do not rely on special attributes of fingerprints it seems suitable to us to base our contribution on fingerprints and should there exists an "on-the-fly DNA-recognition" (or any other suitable biometric identification method) some day which could be embedded into smartcards, the system could easily be switched.

Since we do not want to store templates on the card, we rely on *fuzzy extractors* proposed by DODIS, REYZIN and SMITH. Fuzzy extractors provide the same output, even if the input changes, but remains reasonably close to the original. Since DODIS et al. also claim that their fuzzy extractor's output is nearly distributed uniformly and information-theoretically secure, we translate each user's fingerprint to a unique identifier with it.

Let us assume the underlying anonymous credential protocol is based on the FIAT-SHAMIR identification protocol [FS87, FFS88] or similar to the protocol proposed in [IM03]. Both are Zero-Knowledge-Protocols (ZKP) and have in common, that the prover possesses secret information which he proves to a verifier without revealing to him what that secret information is. If the prover should not be able to share this information, it has to be kept in a tamper-prove device, e.g. a smartcard. Since we want to restrict the use of such a smartcard to a specific person, the secret information is not stored straightly on the smartcard but instead connected to the output of the fuzzy extractor introduced in the previous paragraph. This can be done by a simple modulo addition.

3.1 Creating and Showing Credentials

Before giving an example we briefly overview the setup process of the card and how the credential is showed.

Depending on the underlying ZKP, public parameters and the secret information are determined. The authority stores the secret information s on the card and sets it to “initialisation state”. After reading the user’s fingerprint fp_u , the card computes the value of the fuzzy extractor fe , stores $s^* := s - fe(fp_u)$ and deletes any occurrence of s . Now the card is personalized to a specific user, as the secret information s can only be restored with the value of the fuzzy extractor. To prevent a later change of s^* the card has to be switched to “proving state” at this point.

Proving the possession of a credential is also only slightly changed. The user’s fingerprint is read and the following calculations/proofs are done with $s^* + fe(fp_u)$. It can be easily seen, that this operation restores s . Therefore the underlying ZKP does not need to be changed. We illustrate that by means of an example in the next subsection.

3.2 Example: Modified Feige-Fiat-Shamir Identification Protocol

Setup: The authority chooses two large prime integers p and q and calculates their product $n = pq$. n is then stored on the smartcard and given to the verifier and the prover, p and q are kept secret. Next the authority generates secret numbers s_1, \dots, s_k with $\gcd(s_i, n) = 1$ and computes $v_i \equiv s_i^2 \pmod{n}$. The verifier and prover receive the numbers v_i while the numbers s_i are stored on the card. When the card is initialised to the prover his fingerprint fp_u is read from the card and the stored s_i are overwritten by $s_i^* \equiv s_i - fe(fp_u) \pmod{n}$. The card is set to “proving state” then.

Proving: The smartcard chooses a random integer r , a random sign $\sigma \in \{-1, 1\}$, computes $\sigma x \equiv r^2 \pmod{n}$ and sends this number to the verifier. The verifier chooses numbers $a_i \in \{0, 1\}$ and sends them to the card. The prover now has to give his fingerprint to the smartcard and the card computes $y \equiv r(s_1^* + fe(fp_u))^{a_1}(s_2^* + fe(fp_u))^{a_2} \cdots (s_k^* + fe(fp_u))^{a_k} \pmod{n}$. y is sent to the verifier and he checks if $y^2 \equiv \pm xv_1^{a_1}v_2^{a_2} \cdots v_k^{a_k} \pmod{n}$ to decide if the prover has passed authorisation.

Notice that the user is able to follow the procedure since he is provided with n and v_i and listens to the communication of the card and the verifier to make sure the card follows the protocol.

4 Comparison

Our contribution achieves the same goal as [IM03] to establish an anonymous credential system that makes use of biometrics to obtain non-transferability. Since our approach is very similar, there are almost the same problems and restrictions that apply to [IM03]. To call them by name:

- The security of the system depends very much on the tamper-resistance of the smartcard.
- If a prover begins to interact with a verifier he needs to be kept isolated from the outside world. This is necessary, because otherwise the verifier cannot be sure which card he is communicating with. E.g. if the prover has radio contact to another card, non-transferability suffers and it is possible to share cards.
- The credential is not limited to a specific number of uses. The prover just shows the possession of the secret given by the authority.

However to our feelings there are some improvements to [IM03]:

- The user’s fingerprint is not stored on the smartcard. Although it is not easy to tamper a smartcard, there may be sidechannelattacks like timing attacks [Koc96] or differential power analysis [KJJ99] attacking the concrete implementation of a card. This especially affects match-on-card systems where the fingerprint is read by an external reader and submitted to the card. This way it is a lot easier to create test data and it may be possible to extract information about the stored template on the card if the card is lost.
- Also if the stored secret is revealed it is useless without the information about the fuzzy extractor’s value of the user’s fingerprint, which might be usefull if the card is lost.
- The biometric information is stronger embedded into the anonymous credential system. To us it seems this is a better approach to solve the problem of having to isolate the prover from the outside world. (see next section for more details)

5 Future Work and Open Questions

Our next steps will be to work on a concrete implementation and try to find a way to relax the restriction of isolating the prover from the outside world. It may be possible to have a range of v_i (public information) dependend on the card, which is not suitable for identifying the user but restricts the possibility of “switching to another card” for the user. Another approach may be to elaborate on the possibility of making use of all fingers of the prover to have some kind of input to the card which can’t be easily transmitted to another card. Also we would like to anchor the biometric information deeper in the anonymous credential system and see if it is possible to limit the number of uses similar to [CHK⁺06].

Following the begin of section 2 are there any suitable biometric attributes which can be determined easily and are not able to identify a person? Is there a possibility of reading only parts of a fingerprint so that the user can be sure not being identifiable independet of the technical implementation? And a more technical question: Is it possible to find a procedure that is compatible with the Java Card Biometric API [NIS02] or BioAPI [Bio06]?

References

- [Bio] Biometric Associates, Inc. The BAI Authenticator Smart Card Datasheet. Technical report, <http://www.biometricassociates.com/>.
- [Bio06] BioAPI Consortium. BioAPI Specification Version 2.0 (ISO/IEC 19784-1:2006), 2006.
- [Ble98] Gerrit Bleumer. Biometric yet Privacy Protecting Person Authentication. *Lecture Notes in Computer Science*, 1525:99–110, 1998.
- [CE87] David Chaum and Jan-Hendrik Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. In *Proceedings on Advances in cryptology – CRYPTO ’86*, pages 118 – 167. Springer Verlag, 1987.
- [Cha83] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203, 1983.
- [Cha85] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
- [CHK⁺06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *CCS ’06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 201–210, New York, NY, USA, 2006. ACM Press.
- [CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, volume 2045, pages 93+, 2001.
- [CP93] David Chaum and Torben P. Pedersen. Wallet databases with observers. In *CRYPTO ’92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 89–105, London, UK, 1993. Springer-Verlag.
- [DLN97] Cynthia Dwork, Jeff Lotspiech, and Moni Naor. Digital signets: Self-enforcing protection of digital information. In *Proc. 28th Ann. ACM Symp. on Theory of Computing*, 1997.
- [FDH03] Shaohua Fan, Charles R. Dyer, and Larry Hubbard. Quantification and correction of iris color. Technical Report 1495, 2003.
- [FFS88] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *J. Cryptol.*, 1(2):77–94, 1988.
- [fid08] Homepage of Fidelica Microsystems, Inc., <http://www.fidelica.com/>, January 2008.
- [FS87] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Proceedings on Advances in cryptology—CRYPTO ’86*, pages 186–194, London, UK, 1987. Springer-Verlag.

- [GPR98] Oded Goldreich, Birgit Pfitzmann, and Ronald L. Rivest. Self-delegation with controlled propagation — or — what if you lose your laptop. *Lecture Notes in Computer Science*, 1462:153 – 168, 1998.
- [IM03] Russell Impagliazzo and Sara Miner More. Anonymous Credentials with Biometrically-Enforced Non-Transferability. In *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society (WPES 03)*, pages 60 – 71. ACM, 2003.
- [Jai04] A. K. Jain. Biometric recognition: how do i know who you are? In *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, pages 3 – 5, April 2004.
- [KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. *Lecture Notes in Computer Science*, 1666:388–397, 1999.
- [Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. *Lecture Notes in Computer Science*, 1109:104–113, 1996.
- [LRSW00] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *Selected Areas in Cryptography*, volume 1758. Springer, 2000.
- [NH04] Jonas Nilsson and Michael Harris. Match-on-card for java cards white paper. Technical report, Precise Biometrics, 2004.
- [NIS02] NIST Biometric Consortium – Interoperability, Assurance and Performance Working Group. Java Card Biometric API White Paper, August 2002.
- [Nor04] Björn Nordin. Match-on-card technology white paper. Technical report, Precise Biometrics, 2004.
- [SF04] Richard A. Sturm and Tony N. Frudakis. Eye colour: portals into pigmentation genes and ancestry. *Trends in Genetics*, 20(8):327 – 332, 2004.

Introducing Measurable Path Selection Metrics to Anonymizing Overlay Networks

Johannes Renner
Chair of Computer Science IV
RWTH Aachen University
renner@i4.informatik.rwth-aachen.de

Abstract

Providing anonymity for users of the Internet is a difficult task. Currently, there are only a few systems that are of practical relevance for the provision of low-latency anonymity. One of the most important to mention is Tor which is based on onion routing. Practical client usage of the Tor network though, often leads to delays that are not tolerated by the average end-user. This, in return, discourages many of them from the use of such systems. The latter indirectly lowers the protection for remaining users due to a smaller anonymity set. This work proposes new methods of path selection for the Tor network that are based on measurable performance metrics, instead of self-advertised values. These can be used to improve the average achieved performance, as well as the security of such systems by preventing certain attacks.

1 Introduction

With the growth of the digitized world, privacy issues get more and more importance. Anonymous communication deals with hiding relationships between communicating parties and is a fundamental building block for e.g. privacy-friendly web browsing or identity management systems. Without such protection, attackers are able to deduce information about the network addresses of senders and recipients involved in communications, which is often enough to uniquely identify persons.

Many approaches have already been proposed in order to provide privacy-protection on the network layer, though only some of them have been implemented in praxis (e.g. [DMS04, BFK00]). This work is focused on the most popular and widespread anonymizing system there is today, which is Tor [DMS04], but the proposals can also serve as input for improved designs of future anonymity systems. The Tor network is a circuit switched, low-latency anonymizing overlay network. It is an implementation of the so-called *onion routing* technology that is based on routing TCP *streams* through randomly chosen paths in a network of *onion routers*, while using layered encryption and decryption of the content.

Currently, the publicly accessible Tor network consists of about two thousand servers that are run by volunteers, whereas the number of users is estimated to be hundreds of thousands¹. To learn about the current *network status*, Tor clients (*onion proxies*) download *descriptors* of all currently active routers from a *directory* mirror. A router's descriptor contains all of the necessary information, like its IP address and the used port, as well as self-advertised information regarding its capacities. After choosing a *path* of, per default, 3 routers (*entry*, *middle* and *exit node*), a *circuit* through the network is created using layered encryption in a hop-by-hop manner. Once the circuit is established, it can be used as an anonymizing tunnel for arbitrary TCP connections (for a more detailed description of the Tor design see [DMS04]).

2 Problem Description

The Tor anonymizing network is very dynamic: Everybody can join it by running a server and thus offer available resources for other users. Client usage of Tor though, often leads to significant additional delays caused by the network layers. These delays are mostly due to the inhomogeneity of the network that is caused by its volunteer-based structure, and are often perceived as unnecessary by the users, who then most likely choose to continue surfing the Internet without using Tor. This leads to an overall smaller user base (*anonymity set*), since many users are not willing to sacrifice usability in order to achieve anonymity.

Further, currently used methods of path selection in Tor choose nodes weighted by bandwidth values that are self-advertised by the routers and distributed using the centralized directory service. This fact, however, offers multiple options for attacks, e.g. it is possible to advertise very high bandwidth values to gather as many client paths as possible [BMG⁺07, Dou02]. This work therefore proposes new methods of path selection that can provide an improved performance to the end-users than currently used methods. Further, the proposals will be based on measurable metrics instead of self-advertised values, for being able to prevent certain attacks. The Tor network might then attract more users and hence gain an improved security and anonymity for the single participants [Köp06, DM06].

3 Actively Measuring RTTs

The Tor protocol does currently not provide any mechanisms to measure round-trip times (RTTs) of anonymous communication channels. For sticking with a low latency, it would be helpful to be able to actively measure latencies of Tor circuits, as well as of virtual links between single routers. This proposal is based on an existing prototypical implementation of a method for measuring RTTs in the Tor network that is described in [PPR08]. By making use of Tor's *leaky-pipe* circuit topology, it is possible to extend this technique for measuring RTTs of partial circuits by specifically addressing every single hop of a circuit as the target node once. RTTs of partial circuits can then be used to calculate link-wise RTTs between the single Tor routers in a path containing nodes 0 to n using the following equation:

¹in January 2008

$$RTT_{n-1,n} = RTT_{0,n} - RTT_{0,n-1}$$

For making use of the measured results, it is proposed to model the explored subnet of the Tor network in a graph structure. A so-called *network model* can contain nodes, links between these nodes and arbitrary *node-* or *link-wise* performance metrics. All of the metrics supplied to the model, as well as additional self-advertised information taken from the descriptors, can be used to calculate combined *ranking indices* for either nodes, links or even complete paths. A suitable path that is found in the model is called a *path proposal*. Path selection can then be done probabilistically from the set of all path proposals regarding the ranking indices. This keeps the random aspect in path selection for preserving anonymity, while further being able to control the specific influence of a certain metric on the selection by introducing additional weights.

Besides that, the possibility to measure RTTs on completely established circuits enables us to optimize load-balancing in the Tor network on the circuit layer. This can be done by ensuring in the clients that user streams are always attached to circuits that are currently providing low latencies, using initial or frequent measurements of RTTs on a number of maintained circuits.

4 Estimating Available Bandwidth

Depending on the specific application, a constantly high throughput rate might be more important to a user than latency. Therefore, throughput also needs to be considered as a possible metric for path selection. Instead of weighting the routers by their self-advertised, node-wise bandwidth values from the directory, it is proposed to measure throughput link-wise, for being able to more precisely predict the capacities of specific nodes and links.

Measuring throughput actively by transferring streams over certain nodes for probing their capacities is definitely too much overhead that would have negative impacts on the overall network performance. Therefore, the proposal is to measure throughput passively from within the routers, counting amounts of user-traffic on the single TLS links a node maintains to other nodes on the network, as well as the total amount of traffic it is relaying within a specific time interval. Based on these measurements, routers can give estimations about the currently available capacities on every link to another router. These estimations can regularly be reported to a centralized directory service that averages over all of the information from different sources. This makes it possible to diminish the influence of possible outliers with the aim of detecting, respectively weakening collusions or sybil attacks.

5 Conclusions

The methods that were briefly described in Sections 3 and 4 can most likely be used to improve the performance that is perceived by end-users of Tor regarding the average setup duration,

latency and throughput of the anonymous communication channels. Performance improvements that are gained by these proposals still need to be evaluated though. It will further be necessary to document any possible loss or gain of anonymity that is induced by modifying the used method of path selection.

For a practical adoption of any of the proposed methods, several important design decisions need to be made first. While the use of any link-wise path selection metrics, for example, would allow to more precisely predict the performance of certain paths, the amount of data that needs to be measured, stored in a model and somehow transferred to the clients is quadratically increasing with the number of routers in the network. This will most likely lead to scalability issues in overlay networks with the size of the Tor network.

References

- [BFK00] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 115–129. Springer-Verlag, LNCS 2009, July 2000.
- [BMG⁺07] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against anonymous systems. Technical Report CU-CS-1025-07, University of Colorado at Boulder, February 2007.
- [DM06] Roger Dingledine and Nick Mathewson. Anonymity Loves Company: Usability and the Network Effect. In *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006)*, Cambridge, UK, June 2006.
- [DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.
- [Dou02] John Douceur. The Sybil Attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.
- [Köp06] Stefan Köpsell. Low Latency Anonymous Communication - How Long Are Users Willing to Wait? In Günter Müller, editor, *ETRICS*, volume 3995 of *Lecture Notes in Computer Science*, pages 221–237. Springer, 2006.
- [PPR08] Andriy Panchenko, Lexi Pimenidis, and Johannes Renner. Performance Analysis of Anonymous Communication Channels Provided by Tor. In *Proceedings of the Third International Conference on Availability, Reliability and Security (ARES 2008)*, Barcelona, Spain, March 2008.

A volume-based Accounting System for fixed-route Mix Cascade Systems

Rolf Wendolsky
Universität Regensburg / JonDos GmbH
rolf.wendolsky@jondos.de

Abstract

Strong anonymisation systems currently raise their funds from donations only. A commercial funding, instead, could be an enormous benefit for both anonymisation systems and their end users, as far more money would be available for enhancing the technologies and providing bandwidth. This paper presents a volume-based accounting system for end-user payment of Mix systems with fixed routes as a technical basis for a possible business model for such funding.

1 Introduction

Contemporary strong anonymisation systems rely on volunteers donating servers, bandwidth and development work. However, experience has shown that the number of users quickly outstrips the availability of donations. We therefore looked into ways for commercializing anonymisation systems without compromising security. Ultimately we hope that this may allow such systems to reach a mass market, resulting in more users, more resources for development, independency from donors and thereby better anonymity and security.

Strong anonymization systems are based on so-called Mixes, which should provide unlinkability of incoming and outgoing messages [Chau81], or at least on similar techniques. On the one hand, there are free-route systems, where users may choose any possible combination of Mixes, called circuits, as a path through the network. Hiding the routes is a basic principle of these systems' anonymity. In systems with fixed routes, on the other hand, a user can choose a fixed cascade he wants to use, but cannot select individual Mixes. In this paper, we focus on anonymisation services based on fixed-route Mix cascades.

A flat rate scheme, where no additional data than a ticket is needed to pay the Mix operators of the system, might sound as a very good and easy idea for constructing a payment system for Mix Cascades. However, it has some drawbacks, especially in the fair sharing of income between the participating Mix operators. The details about such a flat rate payment scheme and suggestions for its implementation might be discussed in another paper.

There might also be other incentives for running Mixes, such as offering users tasks for solving, introducing advertisements paid by third parties, or spying on user data. The current paper, instead, focuses on developing a volume-based accounting scheme: users have to pay for the data traffic they cause on Mix cascades. A surrounding business model for the presented technique is also not in the focus of this publication, but may be discussed elsewhere.

We describe requirements for a Mix accounting system in section 2 and discuss related work in section 3. Section 4 shows the commercially available payment methods that can be used to charge end users. Section 5 explains several possibilities for an accounting system. Section 6 summarizes the results and closes the article with some thoughts about possible enhancements.

2 Requirements for secure anonymous accounting in fixed-route systems

Using an accounting system as part of an anonymisation service may sound contradictory, as it requires some kind of user authorization. On the other hand, sender or recipient anonymity are only compromised if it is possible to link sender or recipient to any given message, and relationship anonymity only if sender and recipient can be linked together[PfHa07]. A payment scheme for Mix cascades must therefore ensure that the addition of the payment scheme does not lead to any additional identifying information linked to any message of a given user.

Moreover, the trust models of strong anonymisation systems rely on maximum independence between the operators of the Mixes. If they are bound together by contracts or strong personal relationships, the independence of Mix operators is compromised in both the operational and legal sense. The need for dependencies between the Mix operators due to the payment scheme should therefore be minimized. The fixed cascades itself, however, are not necessary a contradiction to this demand, as each of the Mix operators is free choose any other operator from a pool of possible partners, and is not bound to specific ones.

However, the accounting scheme alone can and should not depict all organisational structures of the whole system. When designing the scheme, one must rely on securely built cascades. This may be achieved, for example, by forcing the operators of the system by contract to build cascades with other independent and locally distributed operators, and not to collude with each other to uncover users. Such contracts must be enforceable by legal means, and could be concluded with the users, the developers of the system or with trustworthy third parties who are guarding the privacy of the system. This model therefore defines Mix cascades as a whole as secure regarding anonymity of users (there are no colluding operators), and focuses on single Mix operators, accounting system operators and outsiders as possible adversaries.

Another important requirement is multilateral security regarding fair accounting for all parties providing services: the payment of Mix operators and the system's developers needs to be assured, and the costs for using the system need to be transparent for end users. It should be hard for these parties to cheat on each other, and cheating parties should at least be caught and banned from the system before serious financial harm can be done.

3 Related work

The payment schemes described in [FrJe98, FrJW98, BaNe99] use anonymous digital cash to pay for the anonymisation service. Basically each Mix packet contains a digital coin as payment for processing that packet. Such schemes provide maximum unlinkability between users, cascades and banks selling the coins. Using anonymous digital cash for real world systems presently fails for at least two reasons: Firstly, there is no bank offering anonymous digital cash. Secondly, practical handling of the digital coins is not easy, especially double spending detection or reimbursement for users who have lost their coins somehow. This article therefore presents solutions which do not depend on the existence of anonymous digital cash.

The Freedom Network developed and operated by Zero Knowledge Systems Inc. was (to the best of our knowledge) the only strong anonymisation system which ever realized an actual working payment system for their service. This accounting system is described in [RuSH00]. One major drawback of it is, that a user has to trust Zero Knowledge not to link his (anonymous) activities to the identity used during payment:

“Another system deficiency is that users must trust Zero-Knowledge to not record any association between activation codes and nym tokens, due to the fact that the system uses untraceable processes rather than blinding cryptographic mechanisms.” [RuSH00]

Our aim is to develop a system which overcomes this limitation, i.e. where the user does not need to trust the service provider. This deficiency, and also the high traffic prices in the past, may have been important reasons for the failure of Freedom besides an oversized business structure.

4 Available payment methods

Although efforts like [ECash] have been made to establish anonymous digital coins for micro-payments, until now no such system is available for real-world use. Therefore other payment methods have to be used for contemporary anonymization systems. These are, for example, credit card, money transfer, cash, prepaid card, e-mail payment and mobile phone payment. Most of these methods require user identification, few of them (prepaid card, cash, money transfer by cash) are really anonymous. Moreover, some of the anonymous ones require a lot of manual processing (cash), have high transaction costs (money transfer by cash) or transaction risks (cash), or are only available in a limited number of countries only (prepaid cards).

An anonymization system should therefore make use of the full spectrum of current payment methods in order to increase its user base as much as possible, while using the most efficient payment methods to maximize both its profit and anonymity. If pseudonymous accounts are used in the payment system, the account numbers should be separated from payment numbers (transaction numbers), in order to keep third parties from linking the users' identities to certain accounts.

5 Volume Package Scheme

5.1 Basic technical ideas

The proposed accounting system has the following attributes:

- All Mix operators are paid according to the traffic usage on their Mixes, regardless of the Mix cascade a Mix is used in.
- End users have to pay for each byte of traffic they cause on any Mix cascade.
- A centralised payment instance does the end user billing and the clearing between end users and Mix operators.
- The payment instance operator may choose the end user prices for traffic volumes independent of Mix operators, and has to pay the Mix operators for their traffic according to Mix-individual prices negotiated beforehand.
- Traffic usage can be proven to users, Mix and payment operators. None of them have any significant ability to cheat by forging traffic numbers.
- End users buy packages of traffic volume at the payment instance, which are stored in pseudonymous user accounts and may then be used as digital payment for Mixes.
- The billing is completely prepaid and neither needs any anonymous, pseudonymous or privacy-friendly payment methods nor any non-cryptographic measures to recognize cheating parties (like plausibility checks). The linkage between user identities and pseudonymous accounts is neither needed nor harmful (except for the general principle of data thriftiness), but may happen depending on the user-chosen payment method. No other organisation than the payment instance provider can link pseudonymous user accounts to specific payments, as account number and transfer number are both random and not linked by any algorithm.
- As long as a pseudonymous user account is being used and not replaced by a new one, the traffic amount that this user consumes on a cascade is logged in the first Mix of the cascade and in the payment instance. The consequences of keeping this information are discussed in section 5.3.

The volume package scheme is most suitable for anonymization systems that do not need extensive or systematic dummy traffic (like all currently operating strong anonymisation systems), as end users would have to pay for this dummy traffic as well. However, since realistic end user prices for data traffic (including the costs of Mix operators and payment operators) are quite low (estimated about 0.5 EuroCent for one MB traffic), this accounting scheme is no big obstacle for users to provide at least some dummy traffic. Better protection against observation, combined with anonymizer clients which send dummy traffic by default, might be enough incentive for users to contribute dummy traffic. As an alternative, dummy traffic could be tagged as such (which restricts its use to protection against outsiders only), and subsequently be ignored for billing.

5.2 Protocol overview

The protocol is explained using the following entities:

- One fixed cascade consisting of several Mixes.
- One payment instance.
- One end user with a client application.

The whole protocol is consistently designed as a prepaid system: users have to create at least one pseudonymous account at the payment instance. This is done by creating a public/private key pair in the users' client, and the payment instance signing the public key. These accounts must then be "charged" with a positive amount of "bytes" before someone can use the anonymizer system. For purchasing these bytes (in the form of volume packages at different price points), any of the payment methods with guaranteed payment from section 4 is suitable. The money transfer has to be finished before any bytes are charged to the account¹, so that there is no need for dunning letters or other forms of debt collection.

When the user connects to a cascade, the client has to confirm a predefined amount of bytes, called *upper limit*, e.g. 3 MegaByte, that he requests to use on this cascade. The client does this by signing an XML document called "cost confirmation" (a digital cheque) that contains the confirmed amount plus all cumulative bytes the user had confirmed for this cascade before, and unambiguous references (by hash values) to the Mixes in this cascade.

Table 1: Cost Confirmation XML elements

Bytes	AccountNumber	PriceCertificates (n*Mix)	Signature
cumulative	random	PriceCertHash + position	account public key

Before the user is allowed to use the cascade, the first Mix cashes in the cost confirmation at the payment instance. If the payment instance replies with a signed confirmation, the user is allowed to proceed. If he reaches a certain amount of bytes lower than the upper limit, called *lower limit*, e.g. 1 MB, the client has to confirm another upper limit of bytes in total, so that the user always keeps a small positive balance of prepaid bytes at the cascade. The first Mix and the client remember the amount of prepaid bytes, even if the client disconnects and reconnects to the cascade, so that connection interruptions do not lead to unnecessary payments, and Mixes cannot cheat the user about the prepaid bytes. Bytes are always counted after protocol package confirmations were received on either side, e.g. after TCP ACK messages.

Even within the cascade itself a prepaid protocol is implemented: all Mixes have virtual accounts at the payment instance. Unlike the user accounts, they do not "contain" bytes, but money. Moreover, the operators of all Mixes have negotiated a price for the traffic going over each Mix with the payment operator. This fixed price is digitally confirmed in so-called "price certificates" that are hard-linked to each Mix' certificate by its hash value.

¹Depending on the payment method, this may take from a few seconds up to some days.

Table 2: Price Certificate XML elements

SubjectKeyIdentifier	Rate	Signature
SKI of Mix certificate	(encrypted) price per GB user traffic	account public key

The price certificates of all Mixes are sent to newly connecting clients, together with the other connection information like Mix certificates, and are part of the cost confirmations sent by the clients (as hashed values). When the first Mix cashes in cost confirmations of users, the payment instance credits as much money to the account of this Mix as corresponds to the confirmed traffic multiplied by the sum of all prices of the price certificates in this cascade. The price certificates are identified inside the cost confirmation by their hash values. The first Mix thus gets the money that he himself is due, plus the money of all other Mixes of the cascade. This Mix in turn has to always confirm a certain amount of bytes (upper Mix limit) to his successor, e.g. 1 GB, by signing *inter-Mix cost confirmations*, before the successor Mix forwards any data traffic from or to this Mix. These inter-Mix cost confirmations contain the price certificates of all successor Mixes and the upper Mix limit plus all cumulative bytes the Mix had confirmed for these successors before.

Table 3: Inter-Mix Cost Confirmation XML elements

Bytes	SubjectKeyIdentifier	PriceCertificates (n*Mix)	Signature
cumulative	Mix SKI	PriceCertHash + position	Mix public key

The successor then pays in the inter-Mix cost confirmations at the payment instance and gets an XML signature for the cost confirmation from the PI, provided the predecessor was able to pay for the traffic. This procedure has to be done between all neighbouring Mixes in the cascade. Like the clients, the predecessor Mixes have to renew the confirmation after a “lower Mix limit” (e.g. 500 MB) has been forwarded from or to them from their successor.

The payment instance gets the users’ money and the proofs (cost confirmations) that this user has spent some bytes using the cascade. The additional proofs shared by the Mixes - the inter-Mix cost confirmations - are not needed by the payment instance. They are only an “insurance” for the middle and last Mix operators against cheating and collaborating first Mixes and the payment instance. This insurance may not be needed as the contracts between payment operator and Mix operators forbids cheating like this, and the payment operator would at least risk to lose his operators if they suspect being fleeced, e.g. by watching their data traffic. In the worst case, this could lead to recourse receivables and other legal actions, so that the organizational security seems sufficient to prevent dishonest behaviour of the payment operator. Instead, the payment instance may take the user cost confirmations, which contain references to all price certificates of a cascade, as a sufficient proof to pay the Mix operators for the traffic confirmed according to their prices. This simplifies the inter-Mix protocol and thereby lowers the possibility of software errors.

The payment operator should calculate an end user price which is guaranteed to be higher than the highest price he has to pay for data traffic on the most expensive cascade. This removes all incentives from the operators to generate data traffic on their own: if they buy volume packages at the payment instance and use their own cascades, they would always lose money. The payment operator therefore has to calculate a maximum price he wants to pay for Mix traffic, and a maximum cascade length, e.g. three Mixes, that he tolerates for payment. The same consideration even shows that it is impossible for the payment operator to offer a volume based flat rate to end users: otherwise, Mix operators had the incentive to buy this flat rate package and generate artificial data traffic on their cascades to maximize their income. This would ruin the payment operator, and could neither be prevented nor recognized.

5.3 Analysis of security aspects

End users have no possibility to use the anonymizer system without paying for traffic because of the system being completely prepaid. Mix operators cannot cheat on the traffic usage, neither against end users (as their clients count the traffic, too), nor against the payment instance (as only cost confirmations for users with a positive balance are accepted). On the other hand, Mix operators have non-ambiguous proofs for the data traffic caused by the users, and for the price they get for it from the payment operator. They may also easily control their costs by arranging a price for the traffic that is higher than their actual traffic costs with the payment operator.

The central collection of user data, more specifically raw data about time and amount of traffic usage on specific cascades, may ease intersection attacks for non-global adversaries ²: If the central payment operator works together with the operators of last Mixes, he may statistically correlate users switching cascades with logical connections outgoing from last Mixes, and may thus get information about the communication partners of such users. If this kind of cooperation with last Mixes exists, the data collection may also permit weak timing attacks. For these reasons, the payment operator should not be allowed to operate any Mixes himself.

On the other hand, accounting by volume forces users to always watch their traffic, and to only connect to the system if needed. Unfortunately, short connection periods generally lower the anonymity of users in the system. The issue may be at least partially addressed by making traffic more deflationary, e.g. by restricting account validity on short time periods of a few months. This also lowers the possibility of Mix and payment operators to keep track of the users's traffic data, as the new accounts cannot be linked to the old ones.

However, some anonymisation techniques that would require lots of dummy traffic may get so expensive, that users had a very strong incentive to connect to the system for single requests only, which would undermine the additional security gained with dummy traffic. Therefore, if dummy traffic should be introduced as security feature, a flat rate scheme would be more suitable. The costs for users would be the same, as Mix operators have to calculate the flat rate prices with non-stop user connections, but there would be no strong incentive for users to switch off the connection to the service any more.

²global adversaries always have this information

6 Conclusion and future work

For fixed-route Mix systems, the proposed volume based accounting is an easy understandable and realisable method to charge the users by Mix and payment operators. For systems that require lots of dummy traffic, it should be replaced or at least complemented by a flat rate scheme following the requirements from section 2 so that the system may provide optimal anonymity. A suitable flat rate scheme may be subject to future work. Moreover, it may be interesting if a volume-based accounting scheme can also be applied on free-route systems. The proposed scheme is not suitable for them, as the price certificates would reveal the path through the Mix network by the signed hashes. More complicated concepts with blind signatures or similar techniques might be applicable, however.

Meanwhile, most elements of the volume package scheme described in this paper have been implemented in a real-world anonymisation system. The basic concepts are working, however there are still no signatures from the payment instance sent to the Mix operators, no cost confirmations between Mixes, and the clients do not check whether any cascade lets them pay more than the upper limit upon reconnection. This productive phase will be used for further studies concerning the feasibility, security and effectiveness of the proposed accounting system.

References

- [BaNe99] Matthias Baumgart, Heike Neumann. Bezahlen von Mix-Netz-Diensten. Verlässliche IT-Systeme – VIS 1999, Vieweg-Verlag, 1999
- [Chau81] David Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM 24/2 (1981) 84–88.
- [ECash] Wikipedia: ECash, 2007. <http://de.wikipedia.org/wiki/ECash>
- [FrJe98] Elke Franz, Anja Jerichow. A Mix-Mediated Anonymity Service and Its Payment. ESORICS '98 (5th European Symposium on Research in Computer Security), Louvain-la-Neuve, LNCS 1485, Springer, Berlin, 1998, 313–327.
- [FrJW98] Elke Franz, Anja Jerichow, Guntram Wicke. Payment Scheme for Mixes Providing Anonymity. IFIP Working Conference on Electronic Commerce 98, LNCS 1402, Springer, Berlin 1998, 94–108.
- [PfHa07] Andreas Pfitzmann, Marit Hansen: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity and Identity Management - A Consolidated Proposal for Terminology, November 2007.
- [RuSH00] Russell Samuels, Ed Hawco: Untracable Nym Creation on the Freedom 2.0 Network. Zero-Knowledge Systems Inc. Whitepaper, 1st November, 2000, <http://osiris.978.org/%7Ebrianr/crypto-research/anon/www.freedom.net/products/whitepapers/Freedom-NymCreation.pdf>

Ein einfaches Anonymisierungsverfahren basierend auf offenen Standards

Benedikt Westermann
Lehrstuhl für Informatik IV
RWTH Aachen, Germany
westermann@i4.informatik.rwth-aachen.de

Zusammenfassung

Alle praktisch existierenden Verfahren zur Anonymisierung im Internet sind wegen ihrer Komplexität nur aufwendig zu realisieren. Weitere Probleme treten aufgrund der Verwendung proprietärer Protokolle auf, da diese als Folge von fehlender Standardisierung öfters geändert werden. All dies erschwert eine parallele Entwicklung alternativer Clients, das Verständnis der Anonymisierung und die mathematische Analyse des Anonymisierungsverfahrens. Das im folgenden beschriebene Verfahren bietet eine Anonymisierung für TCP-Verbindungen, die auf standardisierten Protokollen basiert und durch eine geringe Komplexität und eine gute Verständlichkeit einfach und effizient ist.

1 Einführung

Die Anonymität ist in [PH00] wie folgt definiert:

„Anonymität ist der Zustand, in dem ein Subjekt nicht aus einer Menge von Subjekten, der Anonymitätsmenge, identifiziert werden kann.“

Die Anonymisierung ist der Prozess, der die Anonymität für ein Subjekt schafft. Dies geschieht, indem die Zuordnung zwischen einer Aktion und dem Subjekt aufgehoben wird. Folglich kann nicht mehr unterschieden werden, welches der möglichen Subjekte die Aktion ausgeführt hat.

In unserer Gesellschaft nimmt die Anonymisierung eine bedeutende Rolle ein. Durch die zunehmende Digitalisierung werden Abläufe des Alltags vermehrt über Computer und das Internet abgewickelt, wodurch entsprechende Anonymisierungsverfahren auch im Internet benötigt werden. Ziel dieser Verfahren ist die Anonymisierung auf Verbindungsebene, also das Verbergen der IP-Adressen.

2 Probleme bekannter Anonymisierungsverfahren

Es existieren bisher nur zwei praktisch eingesetzte Verfahren: *Tor* [DMS04] und *JonDonym*¹ [BFK00]. Die anderen in der Literatur vorgestellten Verfahren sind indes theoretischer Natur, zwei Beispiele sind in [RR98, FM02] zu finden.

Tor und JonDonym, die praktisch eingesetzten Verfahren, sind aufwendig zu implementieren, sodass es meistens nur eine einzige Implementierung gibt. Die bei diesem Verfahren verwendeten Protokolle sind einerseits komplex, wodurch eine formale Analyse schwierig ist. Andererseits gibt es ständig Änderungen an den Protokollen. Dies behindert zusammen mit der hohen Komplexität die Entwicklung alternativer Clients sowie das Verständnis der Anonymisierung.

Durch diese Problematik motiviert, wird im Folgenden ein Verfahren vorgestellt, das mit Hilfe des *HTTP*-Protokolls [FGM⁺99] eine Anonymisierung von TCP-Verbindungen bietet, die ähnliche Eigenschaften aufweist wie die von Tor und JonDonym. Der Name des hier vorgestellten Verfahrens ist *Shalon* und steht für: ***Scalable HTTP-based Anonymisation Lightweight Overlay Network***.

Neben der geringen Komplexität und der Verwendung von standardisierten Protokollen bei der Anonymisierung gibt es noch weitere Entwicklungsziele bei Shalon. Das erste ist die dezentrale Verteilung der Server-Informationen, die für den Betrieb Shalons benötigt werden. Ein zweites Entwicklungsziel ist der Schutz gegen lokale Angreifer [DSCP02].

3 Aufbau einer anonymen Verbindung

Die grundlegende Idee des Verfahrens ist, eine TCP-Verbindung über eine Kette mehrerer HTTP-Proxy-Server aufzubauen. Diese Kette wird im Folgenden als *Proxykette* bezeichnet. Ziel der Proxykette ist es, die Verknüpfung zu dem Initiator, also dem Urheber einer Verbindung, und dem Empfänger zu verschleiern. Jedoch reicht eine einfache Verkettung der Proxy-Server allein nicht aus, da der Betreiber des ersten Proxy-Servers ohne eine Verschlüsselung in der Lage ist, den Initiator einer Aktion zuzuordnen.

Eine TCP-Verbindung wird in Shalon wie folgt anonymisiert: Der Initiator baut eine TCP-Verbindung zu dem Proxy-Server P_1 auf. Anschließend wird die Verbindung mit Hilfe von TLS [DA99] verschlüsselt. Im nächsten Schritt wird über die verschlüsselte Verbindung der Connect-Befehl des HTTP-1.1-Protokolls zusammen mit der Adresse des Proxy-Servers P_2 geschickt. Dadurch wird die Verbindung von P_1 zu P_2 verlängert. Alle Daten, die nun in die geöffnete Verbindung geschrieben werden, werden von P_1 an P_2 weitergeleitet. Als nächstes wird die Verbindung zu P_2 verschlüsselt, in dem der Initiator über die aufgebaute Verbindung einen TLS-Handshake initiiert. Proxy P_1 ist durch diese erneute Verschlüsselung nicht mehr in der Lage nachzuvollziehen, zu welchem dritten Proxy sich der Initiator verbindet.

Eine weitere Verlängerung der Verbindung kann erneut, wie oben beschrieben, über den Con-

¹Auch unter dem Namen AN.ON (Anonymität.Online) bekannt

nect-Befehl erreicht werden. Eine auf diese Weise aufgebaute Verbindung ist schematisch in Abbildung 1 dargestellt.

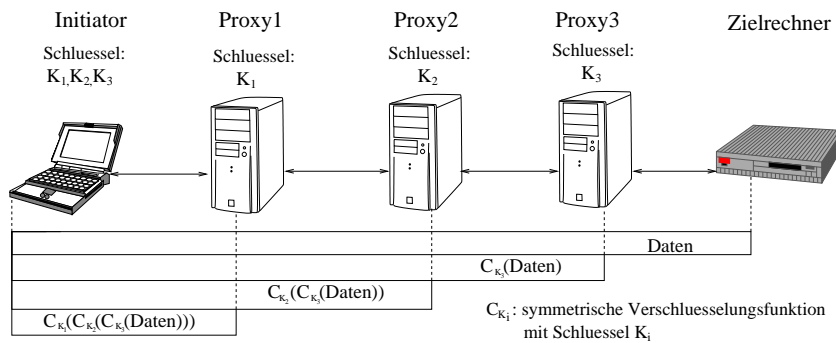


Abbildung 1: Struktur der verschlüsselten Kommunikation

Ist die Verbindung wie in Abbildung 1 aufgebaut, hat diese die gleichen Eigenschaften wie eine aufgebaute TCP-Verbindung. Durch die schichtweise Verschlüsselung ist es einem einzigen Proxy-Betreiber, der die Leitung abhört, nicht möglich eine Verknüpfung zwischen dem Initiator und der Aktion zu erstellen. Für den Zielrechner ist es auch nicht möglich den wahren Initiator anhand der IP-Adresse zu herauszufinden. Die Pfadlänge drei der anonymen Verbindung stellt einen Kompromiss zwischen Sicherheit und Geschwindigkeit dar.

4 Vergleich mit Tor

Die geringe Komplexität Shalons zeigt sich auch bezüglich der Performanz. In Abbildung 2 wird die Transferrate Shalons mit der Transferrate von Tor in Abhängigkeit der Pfadlänge verglichen.

Zur Bestimmung des Datendurchsatzes wurden über eine anonyme Verbindung (Shalon) bzw. über einen Stream (Tor) 50 MB übertragen. Als Server dienten für Shalon sowie für Tor drei Rechner, die jeweils mit zwei Pentium 3 (1 GHz) und 1 GB Arbeitsspeicher ausgerüstet waren. Auf den Servern wurde ein Squid [squ] in der Version 2.6.5 verwendet. Die Tor-Server wurden mit der Tor-Version 0.2.0.12-alpha betrieben. Der Messrechner war indes mit einem Core 2 Duo (2,4 GHz) und 2 GB Arbeitsspeicher ausgestattet.

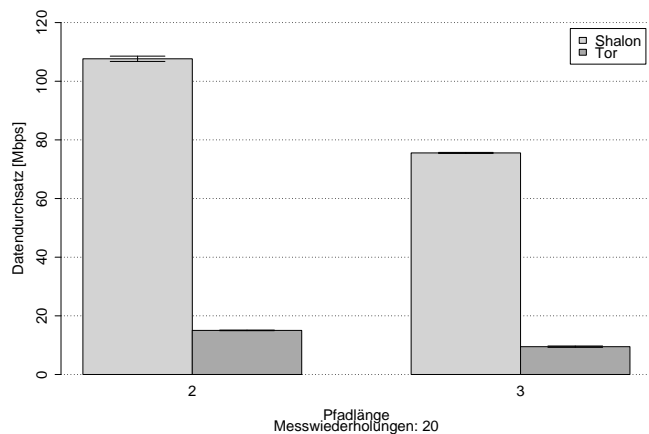


Abbildung 2: Vergleich des Datendurchsatzes

Als symmetrischer Verschlüsselungsalgorithmus wurde bei Shalon der 128-Bit AES Verschlüsselungsalgorithmus verwendet, der auch bei TOR verwendet wird.

Interessant bei diesen Ergebnissen ist der große Unterschied bei dem Datendurchsatz: Shalon ist im Vergleich zu Tor um den Faktor sechs schneller. Der enorme Unterschied zwischen den Transferraten ist u.a. durch das einfachere Protokoll Shalons zu begründen. Bei Tor müssen für die Übertragung von Daten bei einer Pfadlänge von drei mindestens acht Ent-/Verschlüsselungsoperationen durch die Server ausgeführt werden. Im Gegensatz dazu reichen bei Shalon bereits drei Verschlüsselungsoperationen aus, wodurch die Server entlastet werden.

Neben dem Datendurchsatz wurden auch die Latenz- und Aufbauzeiten einer anonymen Verbindung mit Tor verglichen. Die Ergebnisse werden jedoch mit erst mit der vollen Version der Veröffentlichung vorgestellt.

5 Zusammenfassung

Der hier beschriebene Aufbau einer anonymen Verbindung ist im Vergleich zu Tor und Jonym einfacher zu verstehen und zu implementieren, sodass das hier vorgestellte Verfahren durch einen erfahrenden Programmierer innerhalb von einer Woche programmiert werden kann. Durch die geringere Komplexität ist es zudem leichter Fehler zu finden. Messungen der Leistung zeigten außerdem, dass Shalon um einen Faktor sechs schneller ist als Tor. Die Verwendung des HTTP-1.1-Protokolls ermöglicht weiterhin die Wiederverwendung von bereits existierender Software, sodass ausschließlich der Client programmiert werden muss. Als Server kann zum Beispiel der Proxy-Server Squid [squ] verwendet werden.

In der vollen Version der Veröffentlichung wird neben dem hier vorgestellten Verfahren auch eine Sicherheitsanalyse veröffentlicht. In dieser wird diskutiert gegen welchen Angreifer Shalon schützt. Dazu werden bekannte Angriffe gegen Anonymisierungsnetze in Hinblick auf Shalon diskutiert. Im Übrigen wird auch ein Ansatz zur Verteilung der Server-Informationen vorgestellt, der für den Betrieb von Shalon unabkömmlich ist.

Literatur

- [BFK00] BERTHOLD, OLIVER, HANNES FEDERRATH und STEFAN KÖPSELL: *Web MIXes: A system for anonymous and unobservable Internet access*. In: *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, Band 2009/2001, Seiten 115–129. Springer Berlin / Heidelberg, July 2000.
- [DA99] DIERKS, TIM und CHRISTOPHER ALLEN: *The TLS Protocol Version 1.0*. Internet Engineering Task Force: RFC 2246, Januar 1999.
- [DMS04] DINGLELINE, ROGER, NICK MATHEWSON und PAUL SYVERSON: *Tor: The Second-Generation Onion Router*. In: *Proceedings of the 13th USENIX Security Symposium*, August 2004.

- [DSCP02] DÍAZ, CLAUDIA, STEFAAN SEYS, JORIS CLAESSENS und BART PRENEEL: *Towards measuring anonymity*. In: *Privacy Enhancing Technologies 2002*, Band 2482/2003 der Reihe *Lecture Notes in Computer Science*, Seiten 184–188. Springer Berlin / Heidelberg, April 2002.
- [FGM⁺99] FIELDING, ROBERT, JIM GETTYS, JEFF MOGUL, HENRIK FRYSTYK, LARRY MASINTER, PAUL LEACH und TIM BERNERS-LEE: *Hypertext Transfer Protocol: HTTP/1.1*. Internet Engineering Task Force: RFC 2616, June 1999.
- [FM02] FREEDMAN, MICHAEL J. und ROBERT MORRIS: *Tarzan: A Peer-to-Peer Anonymizing Network Layer*. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.
- [PH00] PFITZMANN, ANDREAS und MARIT HANSEN: *Anonymity, Unobservability, and Pseudonymity: A Consolidated Proposal for Terminology*. Draft, July 2000.
- [RR98] REITER, MICHAEL K. und AVIEL D. RUBIN: *Crowds: Anonymity for Web Transactions*. ACM Transactions on Information and System Security, Seiten 66 – 92, April 1998.
- [squ] *Squid*. Internet. <http://www.squid-cache.org> (abgerufen: 26.01.2008).

Bamberger Beiträge zur Wirtschaftsinformatik

Stand Februar 4, 2008

- Nr. 1 (1989) Augsburg W., Bartmann D., Sinz E.J.: Das Bamberger Modell: Der Diplom-Studiengang Wirtschaftsinformatik an der Universität Bamberg (Nachdruck Dez. 1990)
- Nr. 2 (1990) Esswein W.: Definition, Implementierung und Einsatz einer kompatiblen Datenbankschnittstelle für PROLOG
- Nr. 3 (1990) Augsburg W., Rieder H., Schwab J.: Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen
- Nr. 4 (1990) Ferstl O.K., Sinz E.J.: Objektmodellierung betrieblicher Informationsmodelle im Semantischen Objektmodell (SOM) (Nachdruck Nov. 1990)
- Nr. 5 (1990) Ferstl O.K., Sinz E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)
- Nr. 6 (1991) Augsburg W., Rieder H., Schwab J.: Systemtheoretische Repräsentation von Strukturen und Bewertungsfunktionen über zeitabhängigen betrieblichen numerischen Daten
- Nr. 7 (1991) Augsburg W., Rieder H., Schwab J.: Wissensbasiertes, inhaltsorientiertes Retrieval statistischer Daten mit EISREVVU / Ein Verarbeitungsmodell für eine modulare Bewertung von Kennzahlenwerten für den Endanwender
- Nr. 8 (1991) Schwab J.: Ein computergestütztes Modellierungssystem zur Kennzahlenbewertung
- Nr. 9 (1992) Gross H.-P.: Eine semantiktreue Transformation vom Entity-Relationship-Modell in das Strukturierte Entity-Relationship-Modell
- Nr. 10 (1992) Sinz E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM)
- Nr. 11 (1992) Ferstl O.K., Sinz E. J.: Glossar zum Begriffssystem des Semantischen Objektmodells
- Nr. 12 (1992) Sinz E. J., Popp K.M.: Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt
- Nr. 13 (1992) Esswein W., Locarek H.: Objektorientierte Programmierung mit dem Objekt-Rollenmodell
- Nr. 14 (1992) Esswein W.: Das Rollenmodell der Organisation: Die Berücksichtigung aufbauorganisatorische Regelungen in Unternehmensmodellen
- Nr. 15 (1992) Schwab H. J.: EISREVVU-Modellierungssystem. Benutzerhandbuch
- Nr. 16 (1992) Schwab K.: Die Implementierung eines relationalen DBMS nach dem Client/Server-Prinzip
- Nr. 17 (1993) Schwab K.: Konzeption, Entwicklung und Implementierung eines computergestützten Büroorgangssystems zur Modellierung von Vorgangsklassen und Abwicklung und Überwachung von Vorgängen. Dissertation

- Nr. 18 (1993) Ferstl O.K., Sinz E.J.: Der Modellierungsansatz des Semantischen Objektmodells
- Nr. 19 (1994) Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C.: Tool-Based Business Process Modeling Using the SOM Approach
- Nr. 20 (1994) Ferstl O.K., Sinz E.J.: From Business Process Modeling to the Specification of Distributed Business Application Systems - An Object-Oriented Approach -. 1st edition, June 1994
- Ferstl O.K., Sinz E.J. : Multi-Layered Development of Business Process Models and Distributed Business Application Systems - An Object-Oriented Approach -. 2nd edition, November 1994
- Nr. 21 (1994) Ferstl O.K., Sinz E.J.: Der Ansatz des Semantischen Objektmodells zur Modellierung von Geschäftsprozessen
- Nr. 22 (1994) Augsburger W., Schwab K.: Using Formalism and Semi-Formal Constructs for Modeling Information Systems
- Nr. 23 (1994) Ferstl O.K., Hagemann U.: Simulation hierarischer objekt- und transaktionsorientierter Modelle
- Nr. 24 (1994) Sinz E.J.: Das Informationssystem der Universität als Instrument zur zielgerichteten Lenkung von Universitätsprozessen
- Nr. 25 (1994) Wittke M., Mekinic, G.: Kooperierende Informationsräume. Ein Ansatz für verteilte Führungsinformationssysteme
- Nr. 26 (1995) Ferstl O.K., Sinz E.J.: Re-Engineering von Geschäftsprozessen auf der Grundlage des SOM-Ansatzes
- Nr. 27 (1995) Ferstl, O.K., Mannmeusel, Th.: Dezentrale Produktionslenkung. Erscheint in CIM-Management 3/1995
- Nr. 28 (1995) Ludwig, H., Schwab, K.: Integrating cooperation systems: an event-based approach
- Nr. 30 (1995) Augsburger W., Ludwig H., Schwab K.: Koordinationsmethoden und -werkzeuge bei der computergestützten kooperativen Arbeit
- Nr. 31 (1995) Ferstl O.K., Mannmeusel T.: Gestaltung industrieller Geschäftsprozesse
- Nr. 32 (1995) Gunzenhäuser R., Duske A., Ferstl O.K., Ludwig H., Mekinic G., Rieder H., Schwab H.-J., Schwab K., Sinz E.J., Wittke M: Festschrift zum 60. Geburtstag von Walter Augsburger
- Nr. 33 (1995) Sinz, E.J.: Kann das Geschäftsprozeßmodell der Unternehmung das unternehmensweite Datenschema ablösen?
- Nr. 34 (1995) Sinz E.J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme - Entwicklung, aktueller Stand und Trends -
- Nr. 35 (1995) Sinz E.J.: Serviceorientierung der Hochschulverwaltung und ihre Unterstützung durch workflow-orientierte Anwendungssysteme
- Nr. 36 (1996) Ferstl O.K., Sinz, E.J., Amberg M.: Stichwörter zum Fachgebiet Wirtschaftsinformatik. Erscheint in: Broy M., Spaniol O. (Hrsg.): Lexikon Informatik und Kommunikationstechnik, 2. Auflage, VDI-Verlag, Düsseldorf 1996

- Nr. 37 (1996) Ferstl O.K., Sinz E.J.: Flexible Organizations Through Object-oriented and Transaction-oriented Information Systems, July 1996
- Nr. 38 (1996) Ferstl O.K., Schäfer R.: Eine Lernumgebung für die betriebliche Aus- und Weiterbildung on demand, Juli 1996
- Nr. 39 (1996) Hazebrouck J.-P.: Einsatzpotentiale von Fuzzy-Logic im Strategischen Management dargestellt an Fuzzy-System-Konzepten für Portfolio-Ansätze
- Nr. 40 (1997) Sinz E.J.: Architektur betrieblicher Informationssysteme. In: Rechenberg P., Pomberger G. (Hrsg.): Handbuch der Informatik, Hanser-Verlag, München 1997
- Nr. 41 (1997) Sinz E.J.: Analyse und Gestaltung universitärer Geschäftsprozesse und Anwendungssysteme. Angenommen für: Informatik '97. Informatik als Innovationsmotor. 27. Jahrestagung der Gesellschaft für Informatik, Aachen 24.-26.9.1997
- Nr. 42 (1997) Ferstl O.K., Sinz E.J., Hammel C., Schlitt M., Wolf S.: Application Objects – fachliche Bausteine für die Entwicklung komponentenbasierter Anwendungssysteme. Angenommen für: HMD – Theorie und Praxis der Wirtschaftsinformatik. Schwerpunktheft ComponentWare, 1997
- Nr. 43 (1997): Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework - . Accepted for: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1997
- Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using (SOM), 2nd Edition. Appears in: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1998
- Nr. 44 (1997) Ferstl O.K., Schmitz K.: Zur Nutzung von Hypertextkonzepten in Lernumgebungen. In: Conradi H., Kreutz R., Spitzer K. (Hrsg.): CBT in der Medizin – Methoden, Techniken, Anwendungen -. Proceedings zum Workshop in Aachen 6. – 7. Juni 1997. 1. Auflage Aachen: Verlag der Augustinus Buchhandlung
- Nr. 45 (1998) Ferstl O.K.: Datenkommunikation. In. Schulte Ch. (Hrsg.): Lexikon der Logistik, Oldenbourg-Verlag, München 1998
- Nr. 46 (1998) Sinz E.J.: Prozeßgestaltung und Prozeßunterstützung im Prüfungswesen. Erschienen in: Proceedings Workshop „Informationssysteme für das Hochschulmanagement“. Aachen, September 1997
- Nr. 47 (1998) Sinz, E.J., Wismans B.: Das „Elektronische Prüfungsamt“. Erscheint in: Wirtschaftswissenschaftliches Studium WiSt, 1998
- Nr. 48 (1998) Haase, O., Henrich, A.: A Hybrid Representation of Vague Collections for Distributed Object Management Systems. Erscheint in: IEEE Transactions on Knowledge and Data Engineering
- Nr. 49 (1998) Henrich, A.: Applying Document Retrieval Techniques in Software Engineering Environments. In: Proc. International Conference on Database and Expert Systems

- Applications. (DEXA 98), Vienna, Austria, Aug. 98, pp. 240-249, Springer, Lecture Notes in Computer Sciences, No. 1460
- Nr. 50 (1999) Henrich, A., Jamin, S.: On the Optimization of Queries containing Regular Path Expressions. Erscheint in: Proceedings of the Fourth Workshop on Next Generation Information Technologies and Systems (NGITS'99), Zikhron-Yaakov, Israel, July, 1999 (Springer, Lecture Notes)
- Nr. 51 (1999) Haase O., Henrich, A.: A Closed Approach to Vague Collections in Partly Inaccessible Distributed Databases. Erscheint in: Proceedings of the Third East-European Conference on Advances in Databases and Information Systems – ADBIS'99, Maribor, Slovenia, September 1999 (Springer, Lecture Notes in Computer Science)
- Nr. 52 (1999) Sinz E.J., Böhnlein M., Ulbrich-vom Ende A.: Konzeption eines Data Warehouse-Systems für Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule“ im Rahmen der 29. Jahrestagung der Gesellschaft für Informatik, Paderborn, 6. Oktober 1999
- Nr. 53 (1999) Sinz E.J.: Konstruktion von Informationssystemen. Der Beitrag wurde in geringfügig modifizierter Fassung angenommen für: Rechenberg P., Pomberger G. (Hrsg.): Informatik-Handbuch. 2., aktualisierte und erweiterte Auflage, Hanser, München 1999
- Nr. 54 (1999) Herda N., Janson A., Reif M., Schindler T., Augsburg W.: Entwicklung des Intranets SPICE: Erfahrungsbericht einer Praxiskooperation.
- Nr. 55 (2000) Böhnlein M., Ulbrich-vom Ende A.: Grundlagen des Data Warehousing. Modellierung und Architektur
- Nr. 56 (2000) Freitag B, Sinz E.J., Wismans B.: Die informationstechnische Infrastruktur der Virtuellen Hochschule Bayern (vvhb). Angenommen für Workshop "Unternehmen Hochschule 2000" im Rahmen der Jahrestagung der Gesellschaft f. Informatik, Berlin 19. - 22. September 2000
- Nr. 57 (2000) Böhnlein M., Ulbrich-vom Ende A.: Developing Data Warehouse Structures from Business Process Models.
- Nr. 58 (2000) Knobloch B.: Der Data-Mining-Ansatz zur Analyse betriebswirtschaftlicher Daten.
- Nr. 59 (2001) Sinz E.J., Böhnlein M., Plaha M., Ulbrich-vom Ende A.: Architekturkonzept eines verteilten Data-Warehouse-Systems für das Hochschulwesen. Angenommen für: WI-IF 2001, Augsburg, 19.-21. September 2001
- Nr. 60 (2001) Sinz E.J., Wismans B.: Anforderungen an die IV-Infrastruktur von Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule 2001“ im Rahmen der Jahrestagung der Gesellschaft für Informatik, Wien 25. – 28. September 2001

Änderung des Titels der Schriftenreihe *Bamberger Beiträge zur Wirtschaftsinformatik* in *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* ab Nr. 61

Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik

- Nr. 61 (2002) Goré R., Mendler M., de Paiva V. (Hrsg.): Proceedings of the International Workshop on Intuitionistic Modal Logic and Applications (IMLA 2002), Copenhagen, July 2002.
- Nr. 62 (2002) Sinz E.J., Plaha M., Ulbrich-vom Ende A.: Datenschutz und Datensicherheit in einem landesweiten Data-Warehouse-System für das Hochschulwesen. Erscheint in: Beiträge zur Hochschulforschung, Heft 4-2002, Bayerisches Staatsinstitut für Hochschulforschung und Hochschulplanung, München 2002
- Nr. 63 (2005) Aguado, J., Mendler, M.: Constructive Semantics for Instantaneous Reactions
- Nr. 64 (2005) Ferstl, O.K.: Lebenslanges Lernen und virtuelle Lehre: globale und lokale Verbesserungspotenziale. Erschienen in: Kerres, Michael; Keil-Slawik, Reinhard (Hrsg.); Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, S. 247 – 263; Reihe education quality forum, herausgegeben durch das Centrum für eCompetence in Hochschulen NRW, Band 2, Münster/New York/München/Berlin: Waxmann 2005
- Nr. 65 (2006) Schönberger, Andreas: Modelling and Validating Business Collaborations: A Case Study on RosettaNet
- Nr. 66 (2006) Markus Dorsch, Martin Grote, Knut Hildebrandt, Maximilian Röglinger, Matthias Sehr, Christian Wilms, Karsten Loesing, and Guido Wirtz: Concealing Presence Information in Instant Messaging Systems, April 2006
- Nr. 67 (2006) Marco Fischer, Andreas Grünert, Sebastian Hudert, Stefan König, Kira Lenskaya, Gregor Scheithauer, Sven Kaffille, and Guido Wirtz: Decentralized Reputation Management for Cooperating Software Agents in Open Multi-Agent Systems, April 2006
- Nr. 68 (2006) Michael Mendler, Thomas R. Shiple, Gérard Berry: Constructive Circuits and the Exactness of Ternary Simulation
- Nr. 69 (2007) Sebastian Hudert: A Proposal for a Web Services Agreement Negotiation Protocol Framework to be announced
- Nr. 70 (2007) Thomas Meins: Integration eines allgemeinen Service-Centers für PC- und Medientechnik an der Universität Bamberg – Analyse und Realisierungs-Szenarien to be announced
- Nr. 71 (2007) Andreas Grünert: Life-cycle assistance capabilities of cooperating Software Agents for Virtual Enterprises to be announced
- Nr. 72 (2007) Michael Mendler, Gerald Lüttgen: Is Observational Congruence on μ -Expressions Axiomatisable in Equational Horn Logic?
- Nr. 73 (2007) Martin Schissler: to be announced
- Nr. 74 (2007) Sven Kaffille, Karsten Loesing: Open chord version 1.0.4 User's Manual
- Nr. 75 (2008) Karsten Loesing (Hrsg.): Extended Abstracts of the Second Privacy Enhancing Technologies Convention (PET-CON 2008.1)

