Workshop on Freedom and Privacy by Design / CFP2000

Project "Anonymity and Unobservability in the Internet"

Oliver Berthold, Hannes Federrath, Marit Köhntopp

Abstract. It is a hard problem to achieve anonymity for real-time services in the Internet (e.g. Web access). All existing concepts fail when we assume a very strong attacker model (i.e. an attacker is able to observe all communication links). We also show that these attacks are real-world attacks. This paper outlines alternative models which mostly render these attacks useless. Our present work tries to increase the efficiency of these measures.

1 The perfect system

1.1 Attacks

The perfect anonymous communication system has to prevent the following attacks:

1. **Message coding attack:** If messages do not change their coding during transmission they can be linked or traced.
2. **Timing attack:** An opponent can observe the duration of a specific communication by linking its possible endpoints and waiting for a correlation between the creation and/or release event at each possible endpoint.
3. **Message volume attack:** The amount of transmitted data (i.e. the message length) can be observed. Thus, a global observer is able to associate a communication relation to a certain client and server.
4. **Flooding attack:** Each message can only be anonymous in a group of sent messages. All senders of those messages form the anonymity group. Under normal circumstances, each sender should send one message per time interval. However, some of the existing concepts enable an attacker to flood the system in order to separate a certain message.
5. **Intersection attack:** Because of the on-line/off-line periods of the users an attacker may trace them by observation over a long period.
6. **Collusion attack:** A corrupt coalition of users or parts of the system may be able to trace certain users.

Perfect means that there cannot occur a situation where an opponent gets valuable information concerning any communication relation or communication request from and to a certain user. However, no system can protect from an opponent with unlimited power. Therefore we assume that the opponent may not be able to break into cryptographic functions. Though, we have to consider that parts of the anonymous communication system may act as opponents (**insider attacks**).

1.2 Functions of a perfect system

Prevention of collusion attack. A perfect anonymous communication system will be a distributed system. No central system can protect from a corrupt insider since he has all information concerning the sender and recipient of a communication relation. Thus, he can observe a commu-

nication relation. Therefore a distributed system of k (k>1) nodes is supposed. If a maximum of k-1 used nodes ("used" concerning a certain communication relation) are opponents, the protection still works. The first nearly perfect system described in the literature is a network consisting of so-called Mixes [Chaum 1981]. A Mix is a node in a distributed system of k such Mix nodes. A message is transmitted from the sender to the recipient via the k Mix nodes. The opponent may observe all k+1 communication links or k-1 nodes may be corrupt. The operators of the k Mixes should be distinct.

Prevention of message coding attack. To prevent message coding attacks, the sender encrypts the message with k nested layers of public-key encryption. Each Mix removes one layer. Link-to-link encryption between Mixes is not sufficient in order to prevent insider attacks.

Prevention of message volume attack. To prevent message volume attacks, all incoming messages of a Mix have the same length. All outgoing messages of a Mix have the same length. To prevent replay attacks, each message will be processed by a Mix only once.

Prevention of timing attack. It is obvious that a certain message can only be hidden in a group of messages. The communication relation cannot be protected if only one message is transmitted in a certain period of time. Therefore a Mix waits until a defined number n of messages have arrived from n users. Afterwards, all messages will be put out together, but in a different order.

The delay of transmission depends on the behavior of other users, i.e. in situations with low traffic the delay gets high. Since the Chaumian system was designed to protect e-mail, there was no reason to force down the delay. However, if the aim is protecting real-time services (e.g. World Wide Web, chat service etc.), a few modifications and additions to Chaumian Mixes have to be built in.

Dummy messages have to be transmitted in order to reduce delay. Thus, the amount of transmitted messages is likely high enough to process messages immediately. However, it is not sufficient to send dummy messages only between Mixes. Fig. 1 shows an extreme case where dummy traffic between Mixes is completely insufficient because only one message is an ordinary message. Thus, observing in front of the first Mix and behind the last Mix uncovers the communication relation.

Fig. 1: Dummy traffic between Mixes only

Dummy messages have to be sent end-to-end in order to decrease delay as well as maintain the security level.

Prevention of flooding attack. Checking the identity of n users ensures that a single user is not able to flood a Mix with n-1 own messages in order to trace the remaining single message. In a practical system designed for the Internet it is very difficult to prevent flooding attacks since the system has to check the identities. However, in the existing Internet secure identity management is not available. Thus, an opponent can fake different identities in order to simulate different users. It seems to be strange that a secure system has to check the identities of users while they want to remain anonymous. However, we can reduce this demand. In a practical system it has to be ensured that a message is authenticated. Authentication means to check whether a user is per-

mitted to use the service with a specified amount of traffic. It does not mean the identification of users. By means of blinded signatures requests (and messages) can be authenticated without identification of their originator.

Prevention of intersection attack. A perfect system has to prevent intersection attacks: Because of the on-line/off-line periods of the users or a special distinguishable behavior an attacker may trace users by observation over a long period. Ordinary Internet users have a limited number of communication relations and show a very balanced behavior. That means, they have got at most a few hundred e-mail addresses they use, and the number of periodically visited Web sites changes very rarely. More technically spoken, if a client configures his browser to request a certain Web page each time he opens a new browser window, he puts his unobservability at risk. The observer has only to remember the identities of all active users at the time of the request. Later on, when the page is requested again (and again), the observer intersects the previous set of active users with the currently active users. This kind of attack does not definitely uncover a client. However, it dramatically reduces the potential size of the anonymity group. Whenever the user sends individual information (i.e. Cookies, ID numbers, pseudonyms or data of any kind used more than once) that no one else uses, the opponent will be able to uncover all belonging communication relations with a high certainty.

2 Comparison of existing systems

2.1 Why do existing systems fail

With the knowledge of the attacks described above, it is easy to find out the limitations of existing systems. The following systems are well known and some of them can really be used to protect the communication, i.e. some systems are public available: (A comprehensive list of other systems is enclosed at the end of the paper.)

- www.anonymizer.com: Anonymizer is a form-based proxy service. For the following analysis and classification of systems we use Anonymizer as a representative of all systems belonging to that class. Other similar systems are listed at the end of the paper.
- www.research.att.com/projects/crowds: Crowds was proposed by AT&T Research. It is based on secret key cryptography and the concept of a distributed system in order to prevent collusion attacks.
- www.onion-router.net: Onion Routing is a system proposed and maintained by US Naval Research Center. It is based on ideas comparable to the Chaumian Mix concept and provides an anonymous IP-based transport system.
- www.freedom.net: Freedom is a commercial system. It is also based on ideas comparable to Chaumian Mixes and is usable for real-time communication services on the Internet as well. Additionally, Freedom provides a pseudonymous communication infrastructure where users can assume virtual identities, so-called "nyms".

We compare these systems with

1. the classical Mixes introduced in [Chaum 1981] and designed for untraceable e-mail conversation, and
2. our solution for real-time communication (Web Mixes). For more information on our system, see section 4.

See Table 1 for the comparison and the lacks concerning the attacks described above. Whenever a system protects from an attack, the cell in Table 1 is set up in italic type-face.

	Message coding attack	Traffic analysis by means of timing attack	Traffic analysis by means of message volume attack	Traffic analysis by means of intersection attack	Flooding attack	Collusion attack
Anonymizer	*weak protection from outsiders if link encryption between client and proxy; otherwise no protection (insider and outsider)*	no protection	no protection	no protection	no protection or irrelevant*	no protection or irrelevant, because centralized system, proxy knows about communication relation
Crowds	*protects with a certain probability from insiders and in any case from outsiders*	no protection, but request/response bursts are suppressed	no protection	no protection	no protection or irrelevant	*protects with a certain probability; not secure against traffic analysis*
Onion Routing	*protects using public-key cryptography*	no protection at endpoints, protection between Onion Routers	no protection at endpoints, protection between Onion Routers	no protection	no protection or irrelevant	*protects, k-1 of k Onion Routers may collude; not secure against traffic analysis*
Freedom	*protects using public-key cryptography*	no protection at endpoints, however planned: traffic shaping algorithm	no protection at endpoints, however planned: traffic shaping algorithm	no protection	no protection or irrelevant	*protects, k-1 of k Freedom Servers may collude; not secure against traffic analysis*
Chaumian Mixes	*protects using public-key cryptography*	*protects, but high delay*	*protects, but limited and constant msg length*	no protection	no protection or algorithm not feasible on the Internet	*protects, k-1 of k Mixes may collude; not secure against intersection attacks*
Our System (Web Mixes)	*protects using public-key cryptography*	*protects by means of dummy traffic and chop-and-slice algorithm*	*protects by means of dummy traffic and chop-and-slice algorithm*	no protection; Is there actually a solution? Open question: Is combination with blinded message service [Cooper, Birman 1995] useful?	*protects by means of a "ticket"*	*protects, k-1 of k Web Mixes may collude; not secure against intersection attacks*

Table 1: Comparison of systems regarding the general attacks described above
* Remark: If an attack is described as irrelevant, the system already fails due to other attacks that can be launched much easier.

Most practical systems attach great importance to the prevention of message coding attacks and collusion attacks. However, they do not prevent timing attacks, message volume attacks, flooding attacks and intersection attacks. The reason is that these systems were not designed in terms of prevention of these attacks under a very powerful opponent who observes all communication links.

2.2 Qualitative aspects

Table 1 shows a verbal comparison of systems. Now, we try to refine the comparison more qualitatively. Existing concepts for anonymous communication can be classified my means of the items listed in Table 2. Additionally, performance aspects and costs, such as delay and bandwidth efficiency should be taken into consideration.

Good	Better
Systems considering outsider attacks	Systems considering outsider and insider attacks
Systems not considering traffic analysis	Systems considering traffic analysis
Systems with a single point of failure (regarding privacy)	Distributed systems resistant to collusion attacks
Systems protecting from passive attacks (e.g. observing)	Systems protecting from both passive and active attacks (e.g. flooding)

Table 2: Evaluation aspects of systems

Fig. 2 shows a comparison of the systems regarding collusion and strength of observation. In Fig. 3 we compare the systems concerning their delay or bandwidth efficiency. The tradeoff between security (privacy) and efficiency is obvious.

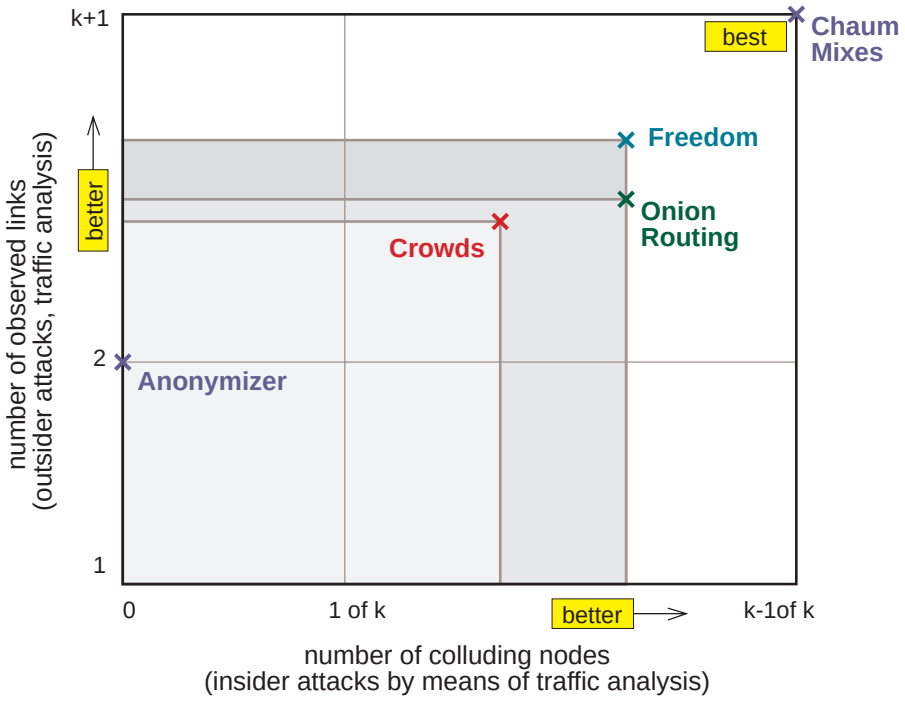

Fig. 2: Comparison regarding protection from collusion and observation

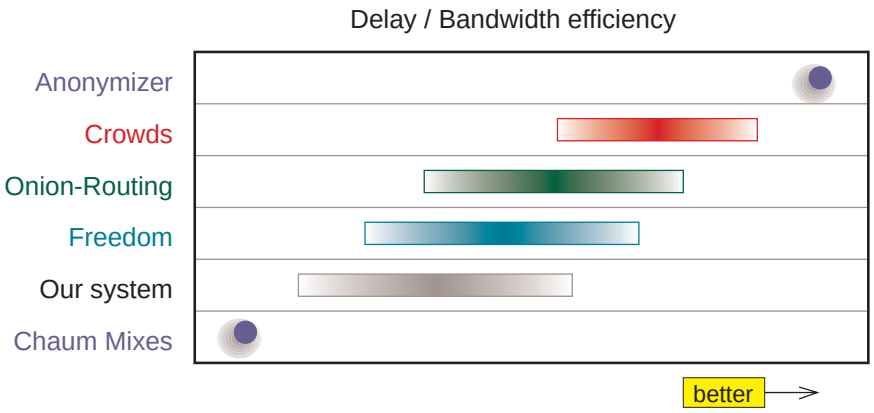

Fig. 3: Comparison regarding efficiency

3 Anonymous services or anonymous networks?

There are two different general approaches to provide anonymity in the Internet. They are:

1. Based on a non-anonymous transport system (layer 1-4 in the OSI reference model), try to implement anonymity in higher layers as far as it is desired.
2. Implement anonymity in the lower OSI layers and build different services upon it with anonymity to a certain extent (e.g. real anonymity, pseudonymity, optional self-identification, authentication, mandatory identification if necessary).

We believe that the second solution, i.e. a purely anonymous network as basis for many kinds of communication, is more suitable because of the following reasons:

- Providers need not know anything about the communication they are routing except for accounting information. It may even be in the interest of the provider not to be able to know anything about the communication to avoid legal liability issues.
- An implementation of anonymity in higher layers based on a non-anonymous transport system is technically much harder (if at all possible) than the other way round.
- With an anonymous transport system, the design of services in higher layers still offers all degrees of freedom.

In particular, the parties involved in an anonymous communication may decide to reveal their identity to the other party or third parties later on in the communication process, i.e. with an anonymous communication network non-anonymous or even authenticated communication is still economical. The opposite may not be true.

Another advantage of a universal solution for anonymity in networks consists in the impossibility to distinguish between the use of different services while observing the transport system.

4 Our System

4.1 Goals

To evaluate the feasibility and costs of anonymity in the Internet and to explore several deployment opportunities we are developing an anonymity system for drug counseling (see

http://www.xtc.mesh.de/.) on the Internet. Our goal is to provide a secure and anonymous technical infrastructure. The special concerns of drug counseling are analyzed by a research group at University Lübeck, Germany. This service will be based on an architecture suitable to provide a purely anonymous network (see section 3) and is therefore usable for any kind of anonymous communication on the Internet.

During the last three years we developed several Mix-based and proxy-based anonymity services (for Web surfing and similar real-time services). Our academical interest is to show that anonymity can be efficiently realized. The special aim is to develop a theoretical background for the efficient implementation of anonymity services in the Internet. We are building an anonymous transport system based on a specific IP format. The goal is to enable asynchronous (like SMTP) as well as nearly synchronous modes of communication (like HTTP) and handle various kinds of packets.

At the moment three implementations or prototypes are available. The first one is a simple form-based proxy service (like Anonymizer.com) that filters dangerous contents (scripts, embedded objects etc.) from the webpage (see http://www.inf.tu-dresden.de/anon/a/). The second one is a Mix-based service using cryptographic functions from Secure Socket Layer (SSL) on both, the client (user) and the server (Mix) side. The third one and newest is a Mix-based implementation that is Java-based on the client side (for platform independent development) while the server side is implemented in C language to achieve high computation speed in cryptographic operations.

The Web site of our project is http://www.inf.tu-dresden.de/~hf2/anon/.

Basically, we use

- a modified Mix concept with
- an adaptive chop-and-slice algorithm (see section 4.2),
- sending of dummy messages whenever an active client has nothing to send,
- a ticket-based authentication system that makes flooding attacks impossible or very expensive and
- a feedback system that gives the user information on his current level of protection.

4.2 Description of functions

Our basic concepts are very similar to other systems based on the idea of Mixes. A Mix scrambles the order of data streams and changes their coding using cryptography to make traffic correlation attacks hard. Constant dummy traffic means that all senders send messages at all times to create the same anonymity groups. If necessary, random data is generated which cannot be distinguished from genuine encrypted traffic. Dummy traffic has to be sent between the endpoints of a communication relation. Dummy traffic only between Mixes is not sufficient to prevent traffic analysis (see section 1.2).

From the Mix model (cf. [Chaum 1981]) we use:

- layered public-key encryption,
- prevention of replay,
- constant message length for incoming and outgoing messages within a certain time period ("slice" see explanation below), and
- changing the order of outgoing messages.

For real-time communication we additionally developed the following concepts both to make traffic analysis harder and to increase the efficiency.

1. Adaptive chop-and-slice algorithm: Large messages (and streaming data) are chopped into short pieces of a specific constant length, called "slice". Each "slice" is transmitted through an anonymous Mix channel. In addition, active users without an active communication request send dummy messages. Thus, nobody knows about the starting time and duration of a communication because all active users start and end their communications at the same time. Otherwise, an observer could determine where and when the anonymous channel starts and ends and find out who is communicating with whom. Dependent on the traffic situation, we modify the throughput and duration of the anonymous channel. The concept of chopping long communications into slices was first introduced in [Pfitzmann et al. 1991]. We use a modified variant with an adaptive duration or throughput.

2. Dummy messages are sent from the the starting point (i.e. client) into the Mix network to make traffic analysis harder (see timing attacks in section 1.2).

3. Ticket-based authentication system for the prevention of flooding attacks: A very difficult problem arises with an active attacker who floods the anonymity service with messages in order to uncover a certain message. This flooding attack is already described in [Chaum 1981]. It had been made harder by the pool concept introduced by Lance Cottrell in Mixmaster (see http://www.obscura.com/~loki/remailer/remailer-essay.html). Although the pool concept was not intended to prevent that attack, it is useful to make the attack much harder for an outsider since it reduces the probability of success. However, insiders (i.e. Mixes) are still able to attack other Mixes with flooded messages. We believe that we found a new concept to suppress flooding of messages both from outsiders and insiders. At first, we limit either the available bandwidth or the number of similar used time slices for each user. Secondly, each user has to show that he is allowed to use the system at the respective time "slice" by providing a **ticket** only valid for the certain "slice". To protect the identity of the user the ticket is a blinded signature issued by the anonymous communication system. More precisely, each Mix issues a limited number of tickets for each "slice" and user. This design of the ticket is useful in order to add the functionality of a prepaid payment system for the anonymity system, too. However, this function has not been implemented yet.

4. Measurement of anonymity level: We believe that it is important for the user to be aware of his level of privacy. It makes a system more reliable and trustworthy for the user. Therefore we analyze the problem how to **measure the anonymity**. However, since the anonymity level depends on the number of active users within the system we need a mechanism or a heuristic that informs the user about his protection level when he requests contents from the Internet. At this time, we are working on the development of reliable and trustworthy mechanisms to visualize the anonymity level.

5. Intersection attacks: As described in section 1.2 an opponent should not be able to do intersection attacks. However, at this time, it is an open question how to prevent intersection attacks. Dummy traffic makes intersection attacks somewhat harder but does not prevent it. We can only give advice to users how they can hinder an observer: The success of an attack can be reduced by trying to avoid sending and receiving linkable events such as sending Cookies, requesting personal Web pages, using pseudonyms in chat services more than once and so on.

5 Concluding remarks

Using Internet services nowadays means leaving digital traces. Anonymity and unobservability on the Internet is a sheer illusion. On the other hand, most people agree that there is a substantial

need for anonymous communication as a fundamental building block of the information society. The availability of anonymous communication is considered a constitutional right in many countries, for example for use in voting or counseling.

The project is a joint venture between the Dresden University of Technology, the Privacy Commissioner Schleswig-Holstein and some providers who are interested in offering anonymity for their users. This constellation of expertise guarantees a high level of technical and jurisdictional knowledge in combination with a hands-on approach, maximizing security and performance.

Compared to Table 1, obviously, the hardest attacks are attacks by traffic analysis, i.e. timing analysis, message volume attacks and intersection attacks. Of course these attacks are hard to launch for an outsider. However, the operator of a system has the possibility to do traffic analysis with a dramatically decreased effort. This is the most important reason why traffic analysis should not give useful information to an attacker. Otherwise, it makes no sense for the client to use Mix-like services (such as Onion Routing, Freedom etc.) if the operator of the service is still able to observe. Simple proxy services with an encrypted link between the client and the proxy would be sufficient. However, the protection of proxies is very limited since proxies do not protect from their operators.

Consequently, the architecture of an anonymity system should properly consider traffic analysis. It is one of the major challenges for the security community to find new and efficient concepts that render traffic analysis useless to improve the users' privacy.

6 References

Remark: Resources mostly available in the Internet (URLs) are not listed here. See Appendix for these resources.

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

David A. Cooper, Kenneth P. Birman: Preserving Privacy in a Network of Mobile Computers. 1995 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos 1995, 26-38.

Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: ISDN-MIXes - Untraceable Communication with Very Small Bandwidth Overhead. 7th IFIP International Conference on Information Security (IFIP/Sec '91), Elsevier, Amsterdam 1991, 245-258.

Appendix

A Related links to our work

- http://www.icsi.berkeley.edu/~hannes/rp.html
- http://www.icsi.berkeley.edu/~hannes/ws.html
- http://www.inf.tu-dresden.de/~hf2/anon/

B Known systems and related projects

Remailers, systems for server anonymity, and systems for pseudonymous e-mail communication are not listed.

B.1 Client anonymity: Simple proxy services

- www.Anonymizer.com — a form-based proxy service
- Aixs.Net/aixs — another form-based proxy service
- Anonymouse.home.pages.de — yet another form-based proxy service by Taker
- www.Rewebber.com — form-based, by Demuth / Rieke (former janus.fernuni-hagen.de)
- www.inf.tu-dresden.de/~hf2/anon/a — Anon Proxy; our implementation of a form-based proxy service
- www.ProxyMate.com / LPWA.com former www.bell-labs.com/project/lpwa Lucent Personalized Web Assistant (LPWA)

B.2 Client anonymity: Services considering attacks via traffic analysis

- www.research.att.com/projects/crowds Crowds by AT&T
- www.onion-router.net by US Navy
- www.freedom.net Freedom by Zero-Knowledge Systems
- www.inf.tu-dresden.de/~hf2/cebit98 Web Mixes; our prototype implementation
- www.KQMLmix.net
- www.privada.net Privada WebIncognito

C Information about the authors

Oliver Berthold, Oliver.Berthold@gmx.de, Phone: ++49 (351) 463-8448, Research Assistant, Dept. of Computer Science, Dresden University of Technology, Germany

Dr. Hannes Federrath, hannes@ICSI.Berkeley.EDU, Phone: ++1 (510) 666-2927, Visiting Research Fellow, International Computer Science Institute (ICSI), Berkeley, CA

Marit Köhntopp, marit@koehntopp.de, Phone: ++49 (431) 988-1214, Head of Department of Privacy-Enhancing Technologies, Privacy Commissioner Schleswig-Holstein, Kiel, Germany