Practical Anonymity for the Masses with Mix-Networks

Marc Rennhard and Bernhard Plattner
Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory, Zurich, Switzerland
{rennhard|plattner}@tik.ee.ethz.ch

Abstract

Designing mix-networks for low-latency applications that offer acceptable performance and provide good resistance against attacks without introducing too much overhead is very difficult. Good performance and small overheads are vital to attract users and to be able to support many of them, because with only a few users, there is no anonymity at all. In this paper, we analyze how well different kinds of mix-networks are suited to provide practical anonymity for a very large number of users.

1 Introduction

Mix-networks [5] are the most promising approach to anonymize communication in the Internet. Originally designed to anonymize e-mail communication, variations of the basic design have led to systems that provide anonymity for low-latency applications such as web browsing. Low-latency mix-networks transport data through the system with at most a few seconds delay, while mix-networks for applications such as e-mail can potentially delay a message in the system for hours. This is the main reason why it is much more difficult to make low-latency mix-networks resistant to an attacker that wants to break the anonymity of the users. In this paper, we focus on low-latency mix-networks, although many results apply to mix-networks in general. Figure 1 depicts the basic idea of a mix-network.

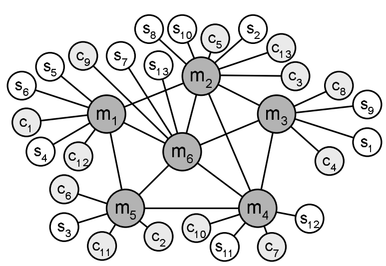

Figure 1. Basic mix-network.

Mix-networks are made up of independent mixes (m_1–m_6) that are distributed in the Internet. Low-latency mix-networks are also called *circuit-based* mix-networks because to access a server (s_i), a client (c_i) chooses a subset of the mixes and establishes a circuit along them. As an example, we assume c_1 communicates with s_1 via m_1, m_6, m_2, and m_3. We name this sequence of mixes c_1's *chain of mixes*. All data sent from c_1 to s_1 are first sent to m_1, then from m_1 to m_6, and so on until the last mix in the chain (m_3) forwards them to s_1. The same chain of mixes is used in opposite order to send data back to c_1. The goal of a mix-network is to make it difficult for an adversary to learn which client c_i communicates with which server s_i. To do so, all messages exchanged between two mixes or clients and the first mix in their chains have the same length, are encrypted or decrypted to change their encoding as they traverse a mix, and are reordered in a mix. In addition, dummy traffic can be used to further complicate the task for an attacker. Traditionally, mix-networks consist of relatively few and well known mixes that are used by a much larger number of users, as shown in figure 1. We name this type *static mix-networks*. Recently, mix-networks where every client is also a mix at the same time have been proposed. Since the mixes in these peer-to-peer based systems can show up and disappear again at any time, we name this type *dynamic mix-networks*.

Mix-networks introduce overhead, which grows with the strength of the threat model the system should be resistant to. Especially the overhead introduced by cover traffic can be huge and keeping the overhead as small as possible is closely related to the anonymity a mix-network provides: a given mix-network with a fixed number of mixes can handle a certain amount of data. If lots of cover traffic is used, less real data can be handled, and fewer people can be supported. This implies that the anonymity set, i.e. the number of people among which one is anonymous, is smaller.

The motivation for this paper is to find the golden mean between the people defining theoretical systems against powerful adversaries that introduce vast amounts of overhead and those implementing practical mix-networks that are often only resistant to weak threat models. Our goal is not to define a system that provides perfect anonymity, and we will point out in section 3 that perfect anonymity in practical mix-networks for low latency applications that support a large number of users is simply not possible. Our goal is to analyze how mix-networks have to be operated to provide practical anonymity for a very large number (e.g. millions) of users. With practical, we mean that the quality of the service a mix-network offers should be good enough such that users actually use the system. We also mean that the number of users a static mix-network can handle per mix must be reasonably large. Finally, with practical we mean that the mix-network should protect from a realistic adversary and not from an extremely powerful, theoretical attacker.

In the next section, we briefly discuss related work. In section 3, we analyze why anonymity in the Internet is such a hard problem and in section 4, we examine the overhead of different cover traffic mechanisms. In section 5, we give arguments for what we call a realistic threat model and in section 6, we discuss how well different approaches to operate mix-networks are suited to support a large number of users. Section 7 concludes our work.

2 Related Work

Several static mix-networks have been operational: Onion Routing [7], Freedom [4], Web Mixes [1], and the Anonymity Network [10]. Onion Routing and Freedom are very similar in their design and do not make use of any cover traffic mechanism. Web Mixes is supposed to defeat a very strong adversary, but until now, no such mechanisms have been included in the prototype. The Anonymity Network employs a relatively efficient cover traffic mechanism, which makes it resistant against certain passive attackers.

The first representative of dynamic mix-networks was Crowds [8], which offered a low level of anonymity. Recent developments include Tarzan [6], which makes use of cover traffic and MorphMix [9], which employs a collusion detection mechanism to detect colluding mixes.

3 Why Anonymity is so Hard

We distinguish between *passive* and *active* attackers. Passive attackers can monitor all or parts of the traffic and try to combine the data observed at various mixes. Active attackers have all the abilities of a passive attacker; in addition they can insert, delete, or modify any data, block links, and control a subset of all mixes. To illustrate the attacks, we use figure 2. One client c_1 is talking to server s_1 via mixes m_1, m_6, and m_3. Similarly, c_2 is communicating with s_2 via m_5, m_6, and m_4. The basic measures employed by the mix-network are fixed-length packets, encryption or decryption of packets as they traverse a mix, and reordering of packets. Consequently, an attacker cannot relate incoming and outgoing packets by looking at their length, encoding, or sequence in which they entered a mix.

The most powerful passive attackers is the global eavesdropper. He can try to correlate incoming and outgoing packets at every mix, as illustrated in figure 2(a, top): c_1 sends three packets to s_1, c_2 two packets to s_2, and the packets happen to arrive at m_6 at nearly the same time. Although an attacker cannot correlate the packets entering and exiting m_6 based on their packet length or encoding, he can still easily deduce that the data from m_1 is forwarded to m_3 and the data from m_5 is forwarded to m_4 because of the different traffic volumes (three packets versus two packets). Using cover traffic that is indistinguishable from the real packets, this attack can be defeated. In Figure 2(a, bottom), m_6 employs constant flows of packets with all its neighbors in both directions. An observer at m_6 cannot tell which of the packets entering and exiting the mix are real ones (the black ones) and which are just dummies (the gray ones). As a result, there is nothing to correlate as the traffic volumes are hidden in the constant flows of packets.

Since the global eavesdropper can observe all mixes in the system, he can also observe all links from clients to the first mix in their chains and from the last mix to the server. In a low-latency mix-network, packets sent from a client to the first mix must exit the system at some other mix within at most a few seconds. Figure 2(b, top) illustrates the attack. The attacker sees three packets entering the mix-network from c_1 and two from c_2. Within some seconds, he sees data exiting at m_3 which has a length corresponding to about three packets and data exiting at m_4 with a length of about two packets. This end-to-end traffic volume attack can still be defeated by employing constant packet streams between the clients and the first mix in their chain, as depicted in figure 2(b, bottom). This removes any correlation between the data entering the first mix and leaving the last mix.

However, more sophisticated attacks are still possible. The long-term intersection attack [2] also correlates events at the endpoints but over a long period of time. It makes use of the fact that the set of users that is connected to the mix-network changes overtime as people leave and come back and that every user has a certain behavior such as visiting regularly the same web servers when being online. Correlating the clients connected to the mix-network and the servers that are contacted at any time allows to break the anonymity of the users if the observation period is long enough. But even this attack could be beaten, at least in theory: by making sure that clients are always connected to the mix-network and always exchange dummy traffic with their first mixes. But this is extremely inefficient in terms of bandwidth usage (section 4) and even if users want to be always online there are offline periods from time to time due to computer or program crash, congested Internet connections, or Internet Service Providers (ISP) failure.

We only look at one type of active attacker: the adversary that controls a subset of the mixes. Dummy traffic can no longer be generated on a per-link basis as is illustrated in figure 2(c, top), where we assume the adversary controls two mixes, m_1 and m_3, which happen to be the first and last

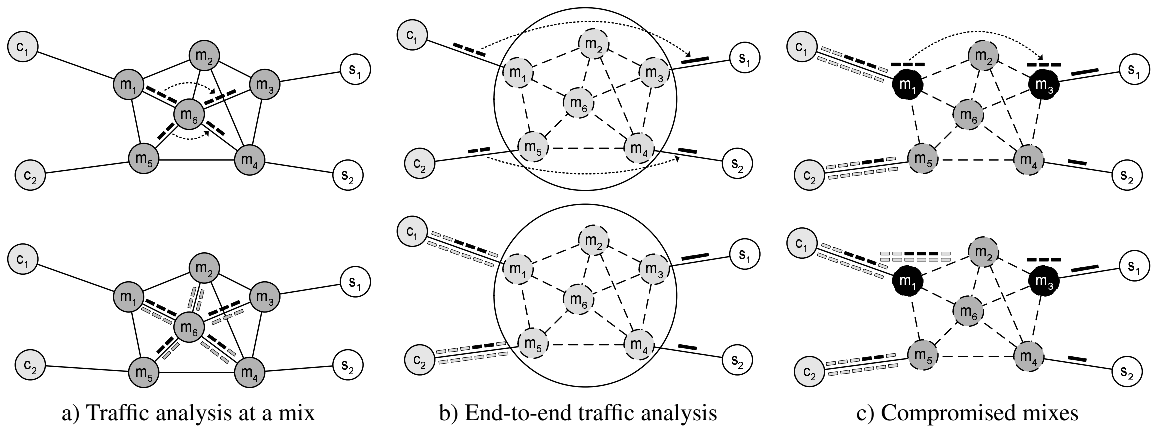

Figure 2. Traffic analysis in mix-networks with (bottom row) and without (top row) cover traffic.

mix used by c_1. Since m_1 knows which packets on the link to c_1 are real data, m_1 and m_3 can carry out an end-to-end traffic volume attack to break c_1's anonymity. To resist this attack, dummies have to be sent from the client through the whole chain of mixes and back, as illustrated in figure 2(c, bottom). As a result, m_1 is no longer able to distinguish between c_1's real data and its dummies and the message volume attack does no longer work.

Unfortunately, the adversary is still not defeated. If m_1 briefly blocks the constant packet stream from c_1 several times and checks with m_3 if it has noted a corresponding brief interruption of an incoming packet stream shortly afterwards, they can conclude with high probability that c_1 communicates with s_1. There are ideas to counter this blocking attack [3] by having the clients include tickets into their packets and a mix only forwards packets after it has accumulated enough of them from different clients. But these ideas are not well applicable to low-latency mix-networks where mixes should forward the data quickly. In addition, they work better with synchronous mix-cascades (special cases of mix-networks where every user uses the same chain of mixes) than asynchronous mix-networks.

We conclude that operating a practical mix-network that supports low-latency applications such that it provides protection against powerful attackers is extremely difficult and may be impossible. A global observer using the long-term intersection attack can most probably beat every system because there are always periods where clients cannot keep up a constant flow of traffic with the first mix in their chain. In addition, an active attacker operating a subset of the mixes can usually find out if it controls the first and last mix in a chain, no matter what cover traffic scheme is used.

4 A Quantitative Analysis of Mix-Networks

Let's assume a mix-network consists of M mixes m_i that are connected to the Internet with bi-directional bandwidths of b_i b/s. We define the *capacity* c of a mix-network as the total number of bits all mixes together can send and receive in a second:

$$c = \sum_{i=1}^{M} m_i \cdot b_i \tag{1}$$

We analyze the minimum capacity a mix-network must offer depending on the number of users. We assume that on average, each client sends d_s bits and receives d_r bits per day through l mixes. So l mixes receive d_s bits in one direction, and l mixes receive d_r bits in the other direction during 24 hours (=86400 seconds) per user. Similarly, l mixes send d_s bits in one direction and l mixes send d_r bits on the way back. On average, each user is responsible that $l \cdot (d_s + d_r)$ bits must be sent and received by the mix-network. The minimum capacity a mix-network must offer to support n users, each of them producing $(d_s + d_r)$ bits during a day can therefore be computed as

$$c_{min} = \frac{n \cdot l \cdot (d_s + d_r)}{86400}. \tag{2}$$

Now we introduce constant bi-directional packet flows on the links between the clients and their first mixes as in figure 2(b, bottom). For simplicity, we do not take the dummies between mixes into account. Like in the case without dummy traffic, each user is responsible that $l \cdot (d_s + d_r)$ bits of real data must be sent and received by the mix-network. But now we also have dummy traffic that is exchanged with the first mix. If u is the average uptime of a client during 24 hours and r_d is the rate at which data are exchanged between the clients and their first mix, then the number of dummy bits received by the first mix is $r_d \cdot u - d_s$. On the way back, the first mix sends $r_d \cdot u - d_r$ to the client in addition to the real data. The minimum capacity is therefore defined as:

$$c_{min} = \frac{n \cdot (l \cdot (d_s + d_r) + r_d \cdot u - \min(d_s, d_r))}{86400} \tag{3}$$

Finally, we evaluate the scheme in figure 2(c, bottom) with end-to-end dummies. The constant packet streams go all the way through the whole chain of mixes and back. Each user is responsible that $r_d \cdot u$ bits are sent to l mixes in the forward direction and to $l-1$ mixes on the way back. In addition, d_r bits are sent from the server to the last mix in the chain. Similarly, $r_d \cdot u$ bits are sent by $2l-1$ mixes and d_s bits are sent by the last mix to the server. As a result, the minimum capacity is defined as:

$$c_{min} = \frac{n \cdot ((2l-1) \cdot r_d \cdot u + \max(d_s, d_r))}{86400} \quad (4)$$

We analyze the impact of cover traffic using a web browsing example. There are 100000 users and each user generates 5 MB of real data per day. We assume there are 0.5 MB web requests and 4.5 MB web replies. The cover traffic rate r_d on the user links is 64 Kb/s. We also distinguish between every client being online for one hour during a day and every client being always online to defeat long-term intersection attacks. Table 1 summarizes the results for the different cases.

Table 1. Mix-networks for 100000 users.

	online time per day (hours)	c_{min} (Mb/s)	dummy overhead
no cover traffic	—	185.2	0%
cover traffic on user link	1	447.2	141%
	24	6580.6	3453%
end-to-end cover traffic	1	1908.3	930%
	24	44841.7	24113%

Dummy traffic significantly increases the minimum capacity. While accepting being vulnerable to long-term intersection attacks introduces an overhead of a "only" a few times the real data, the measures to resist this attack are extremely costly in terms of bandwidth overhead. Note that the figures above were both based on the assumption that all traffic is equally distributed over time. In practice, this is never the case and the effective capacity needed to support 100000 users is probably several times bigger than the minimum capacities we computed above.

Since end-to-end cover traffic is very expensive and since we have seen in section 3 that it does not protect from internal attackers, we conclude that employing end-to-end dummies in practical low-latency mix-networks makes no sense. Employing cover traffic on the user links and between mixes can be more reasonable if passive attackers are a bigger threat than internal attackers.

5 A Realistic Threat Model

We have discussed in section 3 that achieving perfect protection against a global observer or a partial internal attacker in a practical system is simply not possible. But how realistic are such powerful attackers? The community has been arguing for years about what a realistic threat model could be like. In this section, we give arguments for what we call a realistic threat model.

We start with the global passive attacker. If the number of mixes is sufficiently large and they are spread across several countries and ISPs, then the global observer is a very unlikely threat. Looking at a single country such as the US, observing all data is still very difficult. There are currently about 40 backbone ISPs in the US, and the data of all of them must be combined to get the full picture of what's happening. It's unlikely that several ISPs will collude to do so. What about the government? Using FBI's Carnivore diagnostic tool, this is possible by installing Carnivore at all backbone ISPs. But officially, Carnivore can only be used for a limited time after a court order has been issued, and even then only to read data "authorized for capture" by the court order. In addition, a court order is only issued to gather hard evidence and not intelligence. Since a court order is needed for every single temporary installation of Carnivore at an ISP, getting continuous access to all backbone ISPs using the legal way is not likely to be possible for federal agencies. The illegal way would be to convince the backbone ISPs to provide them with all data anyway, which might even work with a few of them. But making deals to collaborate with every single backbone ISP is extremely unlikely to be successful – in particular without anyone leaking information about this criminal act. It is therefore not realistic to assume a global observer if a mix-network consists of several mixes distributed over the whole planet. If a mix-network contains only 10 mixes, then the global attacker is a threat. But with 100s or 1000s of mixes, it is very unlikely an attacker can observe more than, say, 10% of them.

The internal attacker tries to operate mixes by himself. Assume there is quite a big free mix-network consisting of 100 mixes operated by volunteers such as companies, universities, and private persons. Now a government interested in breaking the anonymous communications could operate several mixes by itself. Of course it would not run them under its own name, but provide private persons with the necessary equipment to operate mixes at their homes and pay them 5000$ a year. Assuming the infrastructure (a decent network connectivity and a PC) costs 5000$ a year per mix, there are yearly costs of 10000$ per mix. So if the government manages to convince 300 people to run a mix, they control 75% of all mixes and the yearly costs are 3000000$. Assuming all mixes are equally popular, controlling 75% of all mixes means that the government controls at least the first and last mix in a chain of mixes in $100 \cdot (0.75)^2 = 56.25\%$ of all cases.

We conclude that it is quite possible for an adversary to operate a significant portion (e.g. 50%) of all mixes in a static mix-network operated by volunteers. The larger the number of honest mixes, the more difficult and expensive the attack gets. In particular, in mix-networks made up of volunteers, the threat from an internal attacker controlling a significant portion of the mixes is much bigger than that from an external observer. Since no cover traffic scheme helps against this attacker, the only way to defend against it is to make the attack more expensive by increasing the number of honest mixes.

6 Suitability of Different Approaches

We analyze how well different mix-network approaches are suited to provide anonymity for a large user base. We focus especially on how well the different approaches are suited to acquire enough mixes to support many users and how well they are suited with respect to the realistic threat model we stated in section 5.

6.1 Commercial Static Mix-Networks

The only really big low-latency mix-network was Zero-Knowledge Systems' commercial Freedom network [4]. According to Adam Shostack, Zero-Knowledge Systems' former director of technology, it consisted of about 150 mixes operated by various ISPs in North America, Europe, and Japan. Each mix was connected to the Internet at least at T1 speed (1.544 Mb/s). Assuming double T1 speed on average, this results in a capacity of 463 Mb/s according to (1). No cover traffic was employed and two mixes were used per default in every chain. Assuming that every Freedom user generates 5 MB of data every day, the system was able to support about 500000 users according to (2). Taking overhead and peak times into account, however, Freedom was more likely to support 100000 users with reasonable service, which is about 660 per mix on average.

The Freedom network was certainly big and distributed enough to make the global observer extremely unlikely. In addition, operating several mixes was difficult for a possible internal attacker because Zero-Knowledge Systems made contracts only with ISPs. But a problem with commercial mix-networks is that to sell anonymity, it may not be enough to say "anonymity in 99% of all cases for 50$ a year". Assuming trusted ISPs, cover traffic must be used on the user links and between mixes, and users must be online all the time to offer very strong anonymity. Assuming a dummy data rate of 64 Kb/s on the user links, the theoretical maximum number of users would have dropped to about 7136 according to (3). Taking overheads and peak times into account, something like 1400 users (less than 10 per mix) is more realistic. Considering that Freedom was shut down because of its high cost without employing cover traffic, we can safely state that running a commercial mix-network that employs any kind of dummy traffic is completely out of question – at least in the near future.

6.2 Static Mix-Networks Operated by Volunteers

Leaving out any dummy traffic, table 1 tells that about 200 mixes with a 1Mb/s connection each are needed at minimum to support 100000 users. But how difficult is it to convince 200 independent institutions to operate a mix? There is no easy answer for this question. None of the free mix-networks [7, 1, 10] has grown beyond a limited user trial with at most five mixes, and the mixes were usually not really independent.

What does it cost to run a mix? One must dedicate a reasonably powerful computer and accept that large amounts of traffic are continuously entering and exiting one's network. The first one is not the main problem, but bandwidth is. In Switzerland, you can get a bi-directional 512 Kb/s DSL-connection for about 130$ a month to your home as of March 2003. Not many people are willing to spend this amount of money voluntarily. That leaves universities and large companies, which both have the possibilities to easily spare "a few megabits" of their bandwidth. However, we do not believe the main problem to achieve a critical mass lies in the potential availability of the resources but in the political field. The governments of several countries do not like the idea of anonymity in the Internet, and academic institutions could be threatened to receive less research funding from the government if they operated a mix. Likewise, companies could risk to get bad press about their supporting terrorists and drug dealers and could lose customers. The *Church of Scientology vs. anon.penet.fi* case[1] shows that such threats on operators of anonymity services are not only theoretical. But the fact that collecting a large number of mixes is very difficult means that the internal attacker controlling a large number of mixes becomes a very real threat, and no cover traffic scheme can resist this attacker. One possible defense against an adversary running many mixes is to be very restrictive about who is allowed to operate a mix. But this will make it even more difficult to collect enough mixes to support a large user base.

6.3 Dynamic Mix-Networks

The third option is dynamic mix-networks where every client is also a mix. Every user pays indirectly for the anonymity by dedicating some of his bandwidth and computing power to others, very much like peer-to-peer networks for file-sharing. Since every user brings his own mix, the capacity of a dynamic mix-network grows with the number of users. As a result, a dynamic mix-network can support an infinite number of users in theory.

One problem of static mix-networks is that there are political and legal barriers that may hinder an institution willing to operate a mix from doing so. In a dynamic mix-network, the barrier to join is quite low as in all peer-to-peer

[1] `http://www.xs4all.nl/~kspaink/cos/rnewman/anon/penet.html`

system. Participating in the system knowing that 100000 other users are already using it is a much smaller step than operating one of a small number of static mixes.

Dynamic mix-networks provide good protection from the realistic threats we identified in section 5. With a huge number of mixes distributed all over the world, the probability that any adversary is able to observe a significant portion of the mixes is virtually zero. As a result, no dummy traffic on the user links is needed. In addition, dynamic mix-networks protect much better from internal attacks than static mix-networks, because the only chance to reduce the probability an adversary controls a significant portion of all mixes in a mix-network is to make sure there are very many honest mixes. With every user bringing his own mix, this is the best we can do. In particular, both current dynamic mix-network approaches Tarzan and MorphMix force an adversary to operate his mixes in various subnets. Running 1000 mixes in 1000 different subnets is much more difficult than operating them in one class B network. Consequently, the threat from internal attacks is small.

7 Conclusions

We have presented a thorough discussion about how well mix-networks are suited to provide anonymity for the masses. We have first shown why anonymity is a hard problem. Then we have made a quantitative analysis of mix-networks to show how big these networks must be to support 100000 users depending on the cover traffic scheme they use. We then have presented what we believe is a realistic threat model for mix-networks. Finally, we have examined how well different mix-network approaches are suited to provide practical anonymity for a large user base.

Our goal was to analyze how well mix-networks are suited to provide practical anonymity for the masses. Static mix-networks – operated commercially or based on mixes run by volunteers – seem not to be the right way to go to provide anonymity for the masses. It's questionable if commercial mix-networks can be offered in a profitable way and it's especially arguable if users are willing to pay for "good but not perfect" anonymity. Static mix-networks with mixes operated by volunteers suffer from the problem of acquiring enough mixes and from the very real threat of an internal attacker controlling a significant portion of all mixes, against which no cover traffic scheme can help. Dynamic networks seem to be the best option as they do not suffer from capacity problems and provide good resistance against our realistic threat model without having to employ cover traffic.

Even with very large dynamic mix-networks and assuming our realistic threat model, there is no perfect anonymity. An internal attacker operating some mixes will occasionally break a chain of mixes if he controls both endpoints. Similarly, an adversary observing a few mixes has the chance to break a chain if he eavesdrops on the endpoints. While no cover traffic scheme helps against the first attacker, dummies on the user links and between mixes may reduce the second adversary's chances. But employing cover traffic would reduce the bandwidth available for real data, which would again reduce the practicability of the system.

Dynamic mix-networks are still in their infancy and much more research must be done before more precise statements about their suitability can be made. In particular, since the mixes in these peer-to-peer based systems can disappear at any time, anonymous paths via a subset of the mixes tend to be less stable than in static mix-networks. Nevertheless, we strongly believe dynamic mix-networks are the right choice for the future because they have some vital advantages compared to static mix-networks.

References

[1] O. Berthold, H. Federrath, and M. Köhntopp. Project "Anonymity and Unobservability in the Internet". In *Proceedings of the Workshop on Freedom and Privacy by Design / Conference on Freedom and Privacy 2000 CFP*, pages 57–65, Toronto, Canada, April 4–7 2000.

[2] O. Berthold and H. Langos. Dummy Traffic Against Long Term Intersection Attacks. In *Proceedings of the 2nd Workshop on Privacy-Enhancing Technologies*, San Francisco, CA, USA, April 14–15 2002.

[3] O. Berthold, A. Pfitzmann, and R. Standtke. *The Disadvantages of Free MIX Routes and how to Overcome them*, volume 2009 of *Lecture Notes in Computer Science*. Springer Verlag, 2000.

[4] P. Boucher, A. Shostack, and I. Goldberg. Freedom Systems 2.0 Architecture. White Paper, `http://www.freedom.net/info/whitepapers/Freedom_System_2_Architecture.pdf`, December 18 2000.

[5] D. L. Chaum. Untraceable Electronic Mail, Return Adresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[6] M. J. Freedman and R. Morris. Tarzan: A Peer-to-Peer Anonymizing Network Layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, D.C., USA, November 2002.

[7] M. Reed, P. Syverson, and D. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998.

[8] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, November 1998.

[9] M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In *Proceedings of the Workshop on Privacy in the Electronic Society*, Washington, DC, USA, November 21 2002.

[10] M. Rennhard, S. Rafaeli, L. Mathy, B. Plattner, and D. Hutchison. An Architecture for an Anonymity Network. In *Proceedings of the IEEE 10th Intl. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2001)*, pages 165–170, Boston, USA, June 20–22 2001.