Systems for Anonymous Communication

George Danezis* Claudia Diaz† Paul Syverson‡

August 31, 2009

Abstract

We present an overview of the field of anonymous communications, from its establishment in 1981 by David Chaum to today. Key systems are presented categorized according to their underlying principles: semi-trusted relays, mix systems, remailers, robust & verifiable mixes, and onion routing systems. We include extended discussions of the threat models and usage models that different schemes provide, and the trade-offs between the security properties offered and the communication characteristics different systems support.

Contents

*Microsoft Research, Cambridge, UK.
†K.U.Leuven ESAT/COSIC, Leuven, Belgium.
‡Naval Research Laboratory, Washington DC, USA.

1 Introducing Anonymous Communications

As earlier chapters have shown, electronic cash got its start in the early 1980s in the work of David Chaum. His work in that period introduced primitives that have played a seminal role in research in and development of both electronic cash and financial cryptography in general. These include blind signatures [28], digital pseudonyms and credential mechanisms and their use in electronic payments and other transactions [29], and other primitives and refinements on these.

But even before any of these, he introduced anonymous communication via mixes in 1981 with his seminal paper "Untraceable electronic mail, return addresses, and digital pseudonyms" [27]. A few years later he introduced another of the fundamental primitives used in anonymous communication, the dining cryptographers network, or DC-net [30].

What good is anonymous communication? Anonymous communication hides who is communicating with whom. For example, on the internet anonymous communication would hide a sender's (or recipient's) network address (IP address, email address, etc.) from unwanted observation. What good is this? Obvious commercial uses include browsing for availability or pricing without revealing personal or corporate interest. Similarly one might investigate the offerings of a competitor without identifying oneself. (We know of cases where a vendor has offered a customized website to the known IP address of his competitor that was different from that offered to other addresses.) Individuals hiding their communication profiles also thereby reduce their risk of spear phishing or identity theft.

Anonymous communication is not directly in the design of security for financial and commercial applications, but it is often assumed to be an available building block underlying those applications. Roger Dingledine, designer of the censorship-resistant publishing system FreeHaven and later one of the designers of the anonymous communication system Tor, has remarked that this was a primary motivator that enticed him into researching anonymity. He noticed when working on FreeHaven that, in every censorship-resistant publishing design he encountered, at some point there was a statement that the system

assumes there is an available anonymous communication channel. He wanted to know what was involved in making that assumption correct. Unlike confidential channels and authenticated channels, the history of deployed anonymous channels is rather short. Worse, those concerned with security for financial systems usually understand the need for confidential and authenticated communication, and typically they have at least a rudimentary understanding of what those things are. But, there is often no understanding at all on the part of those concerned with securing financial systems of the principles underlying anonymity nor an understanding of where anonymity might be needed. This chapter is intended to provide some understanding of the design and analysis of systems for communications anonymity.

It might seem that the only issue is to hide the sender from the recipient, but it might be the other way around. It might also be that the sender and recipient wish to be known and even authenticated to each other but don't want their communication to be visible to network observers. Examples include two companies in the early stages of merger discussions or two divisions of a company, for example, if a particular research group suddenly had an unusually large amount of internet communication with the company's patent attorneys in another city. The value of anonymity in all of these cases is to separate identification of communicants from the routing and communication provided by the channel. This separation was first explicitly observed in [89], and its use to support e-cash and similar transactions in [184].

The financial and commercial uses of anonymous communication described above seem obvious now, but interestingly Chaum himself did not suggest the use of mixes for these purposes at the time of their introduction. In general he did not mention applications of his "untraceable electronic mail" in [27] other than to briefly mention voting. (Also note that this was more than a decade before the Web. So some of these uses would be relatively unimaginable, and what was imagined nonetheless is quite remarkable in this context.) He did describe the use DC-nets in [29] to support untraceability of communication underlying transactions.

We will focus in this chapter on anonymous communication channels in the above sense. We will not discuss anonymous, pseudonymous, or generally privacy-preserving transactions for payment, auctions, voting or other purposes that might occur over them. Some of these are covered elsewhere in this book. We will also limit ourselves to the sorts of channels that arise in communications networks. When dealing in purely digital goods and services one can think about anonymizing just at the telecommunications level. Once physical goods are involved, it becomes trickier. How does one take delivery of physical objects without revealing one's address? At least one company attempted to solve this problem as well. In the late 1990s a company called iPrivacy LLC provided privacy protecting commerce for individuals. Besides personalized but anonymized offers and other privacy preserving transactions, intended services included street address obfuscation that delivery company software would only translate when reaching the local postal area. Alternatively a recipient could take "depot delivery", in which she would go to a local delivery depot to re-

ceive the goods by showing an appropriate anonymous identifier. The iPrivacy services were intended to be value adds provided through credit card companies. They also stated that they had arrangements with the U.S. Postal Service to coordinate their delivery system. While an ambitious and interesting idea, iPrivacy was perhaps too ambitious or ahead of its time. Not quite ten years later one must search pointedly for it on the web to find any evidence that it existed. We could find no existing details about the company, its services or their design. In any case, such concerns are beyond our scope.

Since 1981, a body of research has concentrated on building, analyzing and attacking anonymous communication systems. In this survey we look at definitions of anonymous communications and the major anonymous communication systems grouped in families according to the key design decisions on which they are based.

Data communication networks use addresses to perform routing which are, as a rule, visible to anyone observing the network. Often addresses (such as IP addresses, or Ethernet MACs) are unique identifiers appearing in all communication of a user and linking all of the user's transactions. Furthermore these persisting addresses can be linked to physical persons, significantly compromising their privacy.

Anonymizing the communication layer is thus a necessary measure to protect the privacy of users, and to protect computer systems against traffic analysis. Anonymizing communication also supports anonymization techniques at the application layer, such as anonymous credentials, elections and anonymous cash.

In the remaining of this introductory section, we set out the terminology of anonymity properties, we present the existing models for quantifying anonymity, and we explain some limits on anonymity imposed by black box attacks. Section 2 presents anonymity systems based on (centralized) trusted and semi-trusted relays, and introduces the link between anonymous communications and censorship resistance. Mix-based anonymous communication systems are extensively described in section 3. In section 4 we describe re-encryption mixes and other approaches to tracking or proving that mixes process messages properly. Onion rouing systems are described in Section 5. Section 6 introduces other proposals for anonymous communication, and Section 7 presents the conclusions of this survey.

1.1 Terminology

Prior to the quantification of anonymity, a set of working definitions for *anonymity* and other related concepts, such as *unlinkability* or *unobservability* were needed.

In [157], Pfitzmann and Hansen[1] proposed a set of working definitions for anonymity, unlinkability, unobservability and pseudonymity. These definitions have since been adopted in most of the anonymity literature. Their authors continue releasing regular updates on the document addressing feedback from

[1] Hansen was named 'Köhntopp' at the time [157] was published.

the research community[2]. There have been several other papers setting out classifications of communication anonymity, some earlier others later, many using some formal language with an associated logic or formal method for analysis [100, 108, 159, 166, 185], or inspired on the indistinguishability-based formalization of semantically secure encryption, such as the definitions in [105]. These vary in their compatibility with the conceptualization and terminology of Pfitzmann and Hansen as well as in relation to each other. Quotations about terminology below are from Pfitzmann and Hansen unless otherwise noted. We make no attempt to express ideas in a formal language; although we of course intend to be rigorous.

Anonymity. To enable the anonymity of a subject, there always has to be an appropriate set of subjects with potentially the same attributes. Anonymity is thus defined as *the state of being not identifiable within a set of subjects, the anonymity set.*

The *anonymity set* is the set of all possible subjects. With respect to acting entities, the anonymity set consists of the subjects who might cause an action. With respect to addressees, the anonymity set consists of the subjects who might be addressed. Both anonymity sets may be disjoint, be the same, or they may overlap. The anonymity sets may vary over time.

According to the Pfitzmann-Hansen definition of anonymity, the subjects who may be related to an anonymous transaction constitute the anonymity set for that particular transaction. A subject carries on the transaction *anonymously* if he cannot be adequately distinguished (by an adversary) from other subjects. This definition of anonymity captures the probabilistic information often obtained by adversaries trying to identify anonymous subjects.

Unlinkablity. The [ISO15408 1999] defines unlinkability as follows:

"[Unlinkability] ensures that a user may make multiple uses of resources or services without others being able to link these uses together. [...] Unlinkability requires that users and/or subjects are unable to determine whether the same user caused certain specific operations in the system."

We may differentiate between "absolute unlinkability" (as in the given ISO definition above; i.e., "no determination of a link between uses") and "relative unlinkability" (i.e., "no change of knowledge about a link between uses"), where "relative unlinkability" could be defined as follows:

Unlinkability of two or more Items Of Interest (IOIs; e.g., subjects, messages, events, actions, ...) means that within the system (comprising these and possibly other items), from the attacker's perspective, these items of interest are no more and no less related after his observation than they were related concerning his a-priori knowledge.

This means that the probability of those items being related from the attacker's perspective stays the same before (a-priori knowledge) and after the attacker's observation (a-posteriori knowledge of the attacker). Roughly speaking,

[2] `http://dud.inf.tu-dresden.de/Anon\_Terminology.shtml`

providing relative unlinkability of items means that the ability of the attacker to relate these items does not increase by observing the system.

Unobservability. In contrast to anonymity and unlinkability, where not the IOI, but only its relationship to IDs or other IOIs is protected, for unobservability, the IOIs are protected as such. *Unobservability is the state of items of interest (IOIs) being indistinguishable from any IOI (of the same type) at all.*

This means that messages are not discernible from "random noise". As we had anonymity sets of subjects with respect to anonymity, we have unobservability sets of subjects with respect to unobservability. Sender unobservability then means that it is not noticeable whether any sender within the unobservability set sends. Recipient unobservability then means that it is not noticeable whether any recipient within the unobservability set receives. Relationship unobservability then means that it is not noticeable whether anything is sent out of a set of could-be senders to a set of could-be recipients. In other words, it is not noticeable whether within the relationship unobservability set of all possible sender-recipient-pairs, a message is exchanged in any relationship.

Pseudonymity. Pseudonyms are identifiers of subjects. We can generalize pseudonyms to be identifiers of sets of subjects. The subject which the pseudonym refers to is the holder of the pseudonym.

Being pseudonymous is the state of using a pseudonym as ID.

We assume that each pseudonym refers to exactly one holder, invariant over time and not transferable between subjects. Specific kinds of pseudonyms may extend this setting: a group pseudonym refers to a set of holders; a transferable pseudonym can be transferred from one holder to another subject becoming its holder. Such a group pseudonym may induce an anonymity set: Using the information provided by the pseudonym only, an attacker cannot decide whether an action was performed by a specific person within the set [185].

Defining the process of preparing for the use of pseudonyms, e.g., by establishing certain rules how to identify holders of pseudonyms, leads to the more general notion of pseudonymity:

Pseudonymity is the use of pseudonyms as IDs.

An advantage of pseudonymity technologies is that accountability for misbehavior can be enforced. Also, persistent pseudonyms allow their owners to build a pseudonymous reputation over time.

1.2 Anonymity Metrics

Most attacks on anonymous communication networks provide the adversary with probabilistic information about the identity of the entities communicating with each other. This is the reason why information-theoretic anonymity metrics [57, 172] have been widely adopted to quantify the anonymity provided by a variety of designs.

But before information-theoretic anonymity metrics were proposed, there had been some attempts to quantify anonymity in communication networks.

Reiter and Rubin define the *degree of anonymity* as an "informal continuum" from "absolute privacy" to "provably exposed" [166]. Later authors use "degree" to refer to a shifting numerical quantity. Reiter and Rubin, however, do not, which gives them a useful flexibility of expression.

For example, assume for convenience that it is sender anonymity that is of concern. If degree is considered to be, e.g., $1 - p$, where p is the probability that the attacker assigns to potential senders, users are more anonymous as they appear (towards a certain adversary) to be less likely of having sent a message. The metric is very intuitive, but limited in its ability to express the information available to the adversary. Consider two systems each with a hundred users. In the first system, all users, u_i, appear (to the adversary) to be the sender of a message with probability .01. In the second system u_1, Alice, still has probability .01 of being the sender, u_2, Bob, has probability .598 of being the sender, and everyone else has probability .004 of being the sender. In the first system, both Alice and Bob are "beyond suspicion" on the Reiter-Rubin informal continuum, because neither of them is more likely than anybody else to have been the sender. In the second system, Bob has only "possible innocence" because there is a nontrivial probability that someone else is the sender. Everyone else, including Alice, has at least "probable innocence" because each is no more likely to have sent the message than to not have sent the message.

Note that even though Alice's probability of sending is the same in the two systems, in the second one she is no longer beyond suspicion because she is more likely than the other 98 users to be the sender. This is true whether or not Bob happens to be even more suspicious than she is. To underscore this point, consider another system in which the adversary is just a miniscule amount less suspicious of one person. Say u_2, Bob, is .00001 less likely to be the sender than in the first system. So, his probability of being the sender is .00999, and everyone else's is $\approx .0101$. This means that none of the others is beyond suspicion anymore, just Bob. They're all merely probably innocent on the Reiter-Rubin informal continuum. Their continuum was a small part of the paper in which they set out the design of Crowds, which we will discuss in section 2.4. As such, it is not necessarily intended to capture all possible aspects of anonymity of arbitrary systems.

Berthold et al. define the *degree of anonymity* as $A = log_2(N)$, where N is the number of users of the system [17]. This metric only depends on the number of users of the system, and therefore does not express the anonymity properties of different systems. The total number N of users may not even be known. Moreover, adversaries may be able to obtain probabilistic information on the set of potential senders, which is not taken into account in this metric. It is possible to have ranges of anonymity and even conditional notions of anonymity without resorting to probabilities. For example, one of the properties Syverson and Stubblebine describe in [185] is that of $(\leq m)$*-suspected implies* $(\geq n)$*-anonymous*, where $n \leq m$. This means that even if the adversary has narrowed

it to at most m suspects, he cannot narrow it down to fewer than n possible senders (for sender anonymity). This is a property that might seem odd until we consider systems that fail to provide it, i.e., any time the adversary can narrow down the suspect pool to a certain number, he can automatically reduce it significantly further. As a simple example, an anonymity system might be vulnerable to exhaustive search by an adversary for search sets below a certain size.

Information-theoretic anonymity metrics were independently proposed in two papers presented at the *2nd Workshop on Privacy Enhancing Technologies*. The basic principle of both metrics is the same. The metric proposed by Serjantov and Danezis in [172] uses entropy as measure of the *effective anonymity set size*. The metric proposed by Diaz et al. in [57] normalizes the entropy to obtain a *degree of anonymity* in the scale $0 \ldots 1$.

The quantification of anonymity is dependent on the adversary considered. The adversary has certain capabilities and deploys attacks in order to gain information and find links between subjects and items of interest. Most of these attacks lead to a distribution of probabilities that assign subjects a certain probability of being linked to the items of interest. In this respect, a clear and detailed formulation of the attack model considered is a required step in measuring the anonymity provided against that attacker.

The information-theoretic concept of entropy [175] provides a measure of the uncertainty of a random variable. Let X be the discrete random variable with probability mass function $p_i = Pr(X = i)$, where i represents each possible value that X may take with probability $p_i > 0$. In this case, each i corresponds to a subject of the anonymity set; i.e., p_i is the probability of subject i being linked to the item of interest.

The entropy describes thus the information (measured in bits) contained in the probability distribution that describes the links between a set of subjects (the anonymity set) and an item of interest. In [172], entropy is proposed as a measure of the effective anonymity set size. If the entropy is normalized by the maximum the system could provide (if it were perfect and leaked no information) for a given number of users, we obtain a degree of anonymity [57] that gives a measure of the anonymity provider's performance.

Anonymity adversaries may sometimes have access to prior or context information (e.g., after deploying the disclosure attacks explained in the next section). Although intuitively it would seem that considering external sources of information (in addition to the information leaked by the anonymous system) must in all cases reduce anonymity, a counter-example was presented in [58], where it was shown that this is because anonymity corresponds to the entropy of a conditional distribution (i.e, the uncertainty of the adversary given a concrete observation), and not to Shannon's *conditional entropy* [175] (which expresses the average loss of anonymity over all possible observations, and requires the probability distribution of all possible observations to be known). The use of Bayesian inference to combine traffic observations with other sources of information was proposed in [59], where the effects of incomplete or erroneous information are also studied.

A combinatorial approach to measuring anonymity was proposed by Edman et al. in [72] and extended in [84]. The adversary model considered in combinatorial approaches in interested in deanonymizing all the relationships between senders and recipients, as opposed to looking for the set of potential communication partners of a target user.

1.3 Limits and black box attacks

No matter how good the anonymity provided by the network, persistent communication between two users will eventually be detected just by observing the edges of the network and correlating the activity at the two ends of the communication. Such *intersection attacks*, are presented in [17]. These attacks try to extract the sender of a stream of messages by intersecting the sender anonymity sets of consecutive messages sent by a user. This attack model is also considered in [4, 119] and [199]. The statistical variant of this attack is the statistical disclosure attack presented in [39, 49, 136], and extended in [45] to exploit information from anonymous replies. Troncoso et al. proposed in [189] an improvement on these attacks that removes the need to make assumptions on users' sending behavior and achieves greater accuracy by using the available information on all users to deanonymize a given target user. The first intersection attack demonstrated in the wild on a deployed anonymity system was presented in [152].

In [198], Wright et al. present a set of attacks that can be performed by a group of subverted network nodes and then analyze the effectiveness of those attacks against various anonymity network designs. For attacks against mix networks, they calculate the number of routes to be chosen between a sender and a receiver before the full path has been entirely populated by subverted nodes. They also examine the effect that fixed or variable length paths have on this probability. They find similar results for attacks against Crowds, onion routing, and DC-nets. In [199], they extend their analysis to considering a subset of network nodes that simply log all traffic, and provide bounds on how quickly an intersection attack can be performed. Despite these studies being set in the frame of particular systems, like DC-nets and Crowds, they in fact explore fundamental limitations for any systems that select trusted parties at random from a larger set of potentially corrupt nodes to provide security.

Wright et al. also introduced the idea of *helper nodes* which are nodes that are typically picked at random but for repeated or permanent use. This puts an individual communicant at risk from exposing more of his traffic to his helper nodes but also means that if the helpers are not hostile, then the relevant traffic will not eventually be exposed proportional to the fraction of hostile nodes in the network. In [152] it was proposed that such nodes could be chosen based on an expectation of trustworthiness rather than at random, but no analysis of how best to do this was given. Researchers are just beginning to study the implications of routing based on trust, when some nodes are trustworthy or expected to be trustworthy to a different degree than others [73, 116, 181].

2 Trusted and semi-trusted relays

We start presenting anonymous communications systems by introducing systems that rely on one central trusted node to provide security. We will see that they provide a varying, but usually low, amount of anonymity protection against traffic analysis and active attacks.

2.1 The Anon.penet.fi relay

Johan Helsingius started running a trusted mail relay, `anon.penet.fi`, providing anonymous and pseudonymous email accounts in 1993. The technical principle behind the service was a table of correspondences between real email addresses and pseudonymous addresses, kept by the server. Email to a pseudonym would be forwarded to the real user. Email from a pseudonym was stripped of all identifying information and forwarded to the recipient. While users receiving from or sending to a pseudonym would not be able to find out the real email address of their anonymous correspondent, it would be trivial for a local passive attacker or the service itself to uncover the correspondence by correlating the timing of incoming and outgoing email traffic.

While protecting against a very weak threat model, the service was finally forced to close down through legal attacks [102]. In 1996 the "Church of Spiritual Technology, Religious Technology Center and New Era Publications International Spa" reported that a user of `anon.penet.fi` sent a message to a newsgroup infringing their copyright. Johan Helsingius, the administrator of `anon.penet.fi`, was asked to reveal the identity of the user concerned. The case put enormous strain on the service and its operator. Reputation attacks were also experienced, when unfounded reports appeared in mainstream newspapers about the service being used to disseminate child pornography [103].

The service finally closed in August 1996 since it could no longer guarantee the anonymity of its users. The closure was quite significant for the privacy and anonymity research community. In the initial judgment the judge had ruled that "a witness cannot refrain from revealing his information in a trial" [104], even though an appeal was lodged on the grounds of privacy rights protected by the Finnish constitution, and the fact that the information might be privileged, as is the case for journalistic material. Ironically, when the sender behind the relevant pseudonym was revealed it turned out to be that of another remailer, the design of which made further tracing infeasible. While in this case no further tracing was possible, the user interface of the further remailer was more complex. A key attractive feature of Penet was its simple interface, and it is likely that most of Penet's users had not configured for such additional protections.

The concept that even non-malicious relay operators could be forced, under legal or other compulsion, to reveal any information they have access to, provided a new twist to the conventional threat models. Honest relays or trusted nodes could under some circumstances be forced to reveal any information they held concerning the origin or destination of a particular communication. Minimizing the information held by trusted parties is therefore not just protecting

their users, but also the services themselves.

2.2 Anonymizer and SafeWeb

Anonymizer[3] is a company set up by Lance Cottrell, also author of the Mixmaster remailer software, that provides anonymous web browsing for subscribed users. The Anonymizer product acts as a web proxy through which all web requests and replies are relayed. The web servers accessed, should therefore not be able to extract any information about the address of the requesting user. Special care is taken to filter out any "active" content, such as javascript or Java applets, that could execute code on the user's machine, and then signal back identifying information. The original software was written by Justin Boyan, beginning in late 1995 when he was a graduate student at Carnegie-Mellon. The software was sold in 1997 to the company that now bears the name 'Anonymizer' [24].[4]

As for `anon.penet.fi`, the anonymity provided depends critically on the integrity of the Anonymizer company and its staff. The service is less vulnerable to legal compulsion attacks, since no long-term logs are needed, logs that could link users with resources accessed. Unlike email, users always initiate web requests, and receive the replies, and all records can be deleted after the request and reply have been processed. Records can be made unavailable to seize just a few seconds after the provision of the anonymous service to the user.

SafeWeb was a company that provided a very similar service to Anonymizer. The two main differences in their initial products, was that SafeWeb allowed the traffic on the link from SafeWeb to the user to be encrypted using SSL [60], and "made safe" active content in pages by using special wrapper functions. Unfortunately their system of wrappers did not resist a set of attacks devised by Martin and Schulman [134]. Simple javascript attacks turned out to be able to extract identifying information from the users. Most anonymizing proxies, including the Anonymizer, now offer link encryption.

In the absence of any padding or mixing, a passive attacker observing the service would also be able to trivially link users with pages accessed, despite the use of SSL. This vulnerability was studied in [19, 31, 36, 106, 180]. This line of research established that an adversary is capable of compiling a library of 'traffic signatures' for all interesting web-pages that may be accessed. The signatures can then be compared with the traffic characteristics of the encrypted SSL connection to infer the page that was accessed.

The key weaknesses come down to the shape of traffic, which is inadequately padded and concealed. Browsers request resources, often HTML pages, that are also associated with additional resources (images, style sheets, ...). These are downloaded through an encrypted link, yet their size is apparent to an observer, and can be used to infer which pages are accessed. There are many variants of this attack: some attempt to build a profile of the web-site pages and guess from that which pages are being accessed while others use these techniques to beat

[3] `http://www.anonymizer.com/`

[4] Though the name is trademarked, it is also common in the literature to refer generically to anything providing an anonymization function as 'an anonymizer'.

naive anonymizing SSL proxies. In the latter case, the attacker has access to the cleartext input streams and he tries to match them to encrypted connections made to the proxy.

Note that latent structure and contextual knowledge are again of great use for extracting information from traffic analysis: in [36], it is assumed that users will mostly follow links between different web resources. A Hidden Markov Model is then used to trace the most likely browsing paths a user may have taken given only the lengths of the resources that can be observed. This provides much faster and more reliable results than considering users that browse at random, or web-sites that have no structure at all.

2.2.1 Censorship resistance

The threat model that SafeWeb wished to protect against was also very different. The company was partly funded by In-Q-Tel to "help Chinese and other foreign Web users get information banned in their own company *(sic)*" [179]. This claim explicitly links anonymous communications with the ability to provide censorship resistance properties. The link has since then become popular, and often anonymity properties are seen as a pre-requisite for allowing censorship resistant publishing and access to resources. No meticulous requirements engineering study has even been performed that proves (or disproves) that claim, and no cost benefit analysis has ever been performed to judge if the technical cost of an anonymity system would justify the benefits in the eyes of those interested in protecting themselves against censorship. Furthermore no details were ever provided, besides hearsay claims, about groups using this particular technology in a hostile environment, and their experiences with it. The latter would be particularly interesting given the known vulnerabilities of the product at the time.

The first paper to make a clear connection between censorship resistant storage and distribution is Anderson's Eternity service [5]. Serjantov has also done some interesting work on how to use strong anonymous communications to provide censorship resistance [170]. The system presented is, for good technical and security reasons, very expensive in terms of communication costs and latency. Peer-to-peer storage and retrieval systems such as Freenet [33], FreeHaven [62] and, more recently, GnuNet [13] also claimed to provide anonymous communications. Attacks against some anonymity properties provided by GnuNet have been found [128]. Since the design of the two of the three mentioned systems that continue to be developed changes frequently and is not always documented, it is very difficult to assess the security or the anonymity they provide at any time. Feamster et al. have looked at different aspects of web censorship resistance by making use of steganography to send high volumes of data, and what they called "URL hopping" to transmit requests [76, 77]. Finally, aChord [101] presents concisely the requirements of a censorship resistant system, and attempts to build one based on a distributed hash table primitive.

Aside from complete systems, many isolated mechanisms have been proposed to bypass restrictive firewalls that attempt to prevent access to an anonymous

communication system. The Tor [65] system, provides a mode that tunnels everything over TCP Port 80 which is often not filtered since it is usually reserved for HTTP (Web) traffic. Anonymizer relies on providing people behind national firewalls (notoriously in China and Iran) with network addresses that are not being filtered because they are unknown to the firewall [131]. This results in an arms race between the providers of fresh addresses, that extensively rely on spam for distribution, and the authorities that seek to detect them and block them. A similar architecture [127], that relies on volunteers donating their non-blocked network address to help those behind filtering firewalls, has been described and implemented by the JAP[5] project. More recently the Tor Project has designed and deployed a similar system for blocking resistance [64]. It employs multiple simultaneous mechanisms for discovery of "bridges" to the Tor network.

Other studies have looked at censorship and Internet filtering in China [192], and the specific capabilities of the national firewall [35]. It was discovered that it simply sends TCP resets to force communicating parties, with compliant TCP/IP stacks, to drop their connections. Modified clients that ignore such resets were able to carry on communicating. Finally, two studies, one German [70] and one British [34], have looked at the effectiveness of Internet Service Providers filtering out web sites that are known to contain child pornography. In their study of the live BT content blocking system *cleanfeed* [34] they discovered that forbidden content could be trivially accessed. Furthermore, the blocking mechanism had precisely the opposite of its intended effect in that it could be used as an oracle for interested parties to discover sites with illegal material.

2.3 Type I "Cypherpunk" remailers

Type I remailers, first developed by Eric Hughes and Hal Finney [155], are nodes that relay electronic mail, after stripping all identifying information and decrypting it with their private keys. The first code-base was posted to the cypherpunks mailing list, which gave the remailers their nickname. The encryption was performed using the Pretty Good Privacy (PGP) public key encryption functions. The encoding scheme was also designed to be performed manually, using standard text and email editing tools. Many remailers could be chained together, in order for users not to rely on a single remailer to protect their anonymity.

Reply blocks were supported to allow for anonymous reply addresses. The email address of the user would be encrypted using the remailer's public key, and inserted in a special header. If a user wished to reply to an anonymous email message, the remailer would decrypt it and forward the contents.

Type I remailers offer better resistance to attacks than the simple `anon.penet.fi` relay. No database that links real user identities with pseudonyms is kept. The addressing information required to reply to messages is included in the messages themselves, in an encrypted form.

[5] `http://anon.inf.tu-dresden.de/`

The encryption used when the messages are forwarded through the network prevents the most trivial passive attacks based on observing the exact bit patterns of incoming messages and linking them to outgoing ones. However it leaks information, such as the size of the messages. Since PGP, beyond compressing the messages, does not make any further attempts to hide their size, it is trivial to follow a message in the network just by observing its length. The reply blocks provided are also a source of insecurity. They can be used many times, and an attacker could encode an arbitrary number of messages in order to mount an attack to find their destination. Since all replies encoded with the same reply block would contain an identical sequence of bytes, this attack is even easier than the statistical disclosure attacks [39, 49]. The attack can then be repeated to trace any number of hops.

Despite these drawbacks, type I remailers became very popular. This is due to the fact that their reply capabilities allowed the use of Nym Servers. Their reply block feature, that is not present in the later type II Mixmaster software, is both essential to build nym servers, but also insecure even against passive adversaries. This has prompted the design of Mixminion, a type III remailer, that is extremely resistant to traffic analysis, and provides secure single-use reply blocks.

2.4 Crowds

Crowds [166] was developed by Reiter and Rubin at AT&T Laboratories. It aims to provide a privacy preserving way of accessing the web, without web sites being able to recognize who is browsing. Each client contacts a central server and receives the list of participants, the "crowd". A client then relays her web request by passing it to another randomly selected node in the crowd. Upon receiving a request each node tosses a biased coin and decides if it should relay it further through the crowd or send it to the final recipient. Finally, the reply is sent back to the initiating client via the path established as the request was being forwarded through the crowd. Both requests and replies along a single path are encrypted using a path key that is sent from the initiating client and passed along to all nodes on that path as it is established.

Crowds is a landmark in anonymity research since its security relies on the adversary not being able to observe the links. Instead, the adversary is assumed to only control a fraction of nodes in each crowd, and the ultimate web server. Although this threat model was initially believed to be unrealistic and fragile, it was later realized that it can be achieved using simple link encryption and padding.

A system which also relies for its security on the inability of the adversary to intercept all communication channels was presented by Katti et al. in [118]. They conceptually 'slice' each message into many parts, using a secret sharing scheme, and send them out in the network using different channels to an intermediary: if the adversary fails to observe one of the channels they cannot reconstruct the message or the final address to which it is destined. The scheme can be simplified, by considering the secret sharing scheme over a partially se-

cure set of channels, as a primitive encryption mechanism, and the intermediary as a trusted relay that is to decrypt the message and forward it.

Crowds is one of the first papers to address quantitatively how colluding nodes would affect the anonymity provided by the system. It is clear that after the first dishonest node in the path of a request no further anonymity is provided because the cleartext of all requests and replies is available to intermediate nodes. Therefore, given a certain fraction of colluding attacker nodes it is possible to measure the anonymity that will be provided [57].

Crowds also introduces the concept of *initiator anonymity*: a node that receives a request cannot know if the previous node was the actual requester or was just passing the request along. This property is quite weak and two independent groups have found attacks that identify the originator of requests [198, 178]. They discovered that if a client repeatedly requests a particular resource, they can eventually be linked: The attack relies on the intuition that if a repeated request is made each time via another randomly formed path, the true initiator of the repeated request will appear on more of those paths than will a random node in the crowd. Thus, if corrupt nodes on different paths that pass the repeated request observe that the same node is the predecessor on those different paths forwarding the request to them, its chances of being the true initiator are higher. Therefore, for each resource accessed it is sufficient to count how many times each node is seen to be accessing it, and select the node corresponding to the most requests as the most likely initiator.

The basic attack notion was known to Reiter and Rubin, and they cite the analysis they present in [166] as the reason that their design has static paths: crowd clients build a single path through the crowd and then use it indefinitely. If a node on the path becomes unreachable, the prior node in that path becomes the last node for that path. Since the path only shortens, there is no selection of new successor nodes, thus no way to conduct the above attack. What Reiter and Rubin did not account for was communication relationships that persist longer than crowds do. To prevent new parties who join a crowd from standing out, all clients must form new paths whenever someone joins. When Crowds was deployed the default setting to manage changes in crowd membership was to reform once a day. This was enough to conduct effective predecessor attacks. Another interesting feature discovered by Shmatikov was that anonymity gets worse as the crowd gets *bigger*. While intuition might say that a bigger crowd makes for a bigger anonymity set, he observed that predecessor attacks became more effective as crowd size increased because the probability that any node other than the initiator might occur on a path went down as the crowd got bigger. This attack sensitized the anonymity community to the problem of protecting *persistent* relationships instead of simple single message or request exchanges.

Despite the difficulty of securing initiator anonymity, a lot of subsequent systems such as achord [101] and MorphMix [168], try to achieve it.

2.5 Nym servers

Nym servers [137] store an anonymous reply block, and map it to a pseudonymous email address. When a message is received for this address it is not stored, but immediately forwarded anonymously using the reply block to the owner of the pseudonym. In other words, Nym Servers act as a gateway between the world of conventional email and the world of anonymous remailers. Since they hold no identifying information, and are simply using anonymous reply blocks for routing purposes, they do not require users to trust them in order to safeguard their anonymity. Over the years, special software has been developed to support complex operations such as encoding anonymous mail to go through many remailers, and managing Nym Server accounts. Mathewson presents a contemporary design for a Nym server called *Underhill* [135] that uses the state of the art in remailer technology, and *NymBaron* is its current implementation [80].

Nym servers are also associated with pseudonymous communications. Since the pseudonymous identity of a user is relatively persistent, it is possible to implement reputation systems, or other abuse prevention measures. For example, a nym user might at first only be allowed to send out a small quantity of email messages. This can be increased over time, as long as abuse reports are not received by the nym server operator. Nym servers and pseudonymous communications offer some hope of combining anonymity and accountability.

At the same time, it is questionable how long the true identity of a pseudonymous user can be hidden. If all messages sent by a user are linked between them by the same pseudonym, one can try to apply author identification techniques to uncover the real identity of the user. Rao and Rohatgi, in their paper entitled "Can Pseudonymity Really Guarantee Privacy?" [163] show that the frequency of function words[6] in the English language can be used in the long term to identify users. A similar analysis could be performed using the sets of correspondents of each nym, to extract information about the user. Mathewson and Dingledine have noted in [136] that statistical disclosure attacks are very effective at linking pseudonyms with their corresponding users, when those are based on remailer systems.

The shortcomings of remailer based systems have prompted a line of research that looks at alternative techniques to provide receiver anonymity. Techniques from Private Information Retrieval (PIR) have been suggested. PIR is concerned with a family of techniques that allow clients to query a database, without the database or any third party being able to infer which record was retrieved. PIR in the context of receiver anonymity can be implemented either using secure hardware [6, 120], or distributed servers [123, 169]. Private Search, a simplified PIR construction [18, 44, 151] has the potential to be used in efficient receiver anonymity systems.

Interestingly, Ishai et al. also show that given a strong anonymous channel, one can construct an efficient Private Information Retrieval system [109].

[6] Function words are specific English words used to convey ideas, yet their usage is believed to be independent of the ideas being conveyed. For example: *a, enough, how, if, our, the, ...*

3 Mix systems

The type I remailer, presented in section 2.3, is a relatively simple and relatively insecure version of a mix system. There is a large body of research on mix systems and mix networks. This section presents secure constructions based on these ideas.

3.1 Chaum's original mix

The first, and most influential, paper in the field of anonymous communications was [27]. Chaum introduced the concept of a "mix" node that hides the correspondences between its input messages and its output messages in a cryptographically strong way.

The work was done in the late seventies, when RSA public key encryption was relatively new. For this reason the paper might surprise today's reader by its use of raw RSA, the direct application of modular exponentiation for encryption and decryption, along with an ad-hoc randomisation scheme. Nonces are appended to the plaintext before encryption in order to make two different encryptions output different ciphertext.

The principal idea is that messages to be anonymized are relayed through a node, called a mix. The mix has a well-known RSA public key, and messages are divided into blocks and encrypted using this key. The first few blocks are conceptually the "header" of the message, and contain the address of the next mix. Upon receiving a message, a mix decrypts all the blocks, strips out the first block that contains the address of the recipient, and appends a block of random bits (the junk) at the end of the message. The length of the junk is chosen to make message size invariant. The most important property that the decryption and the padding aim to achieve is *bitwise unlinkability*. An observer, or an active attacker, should not be able to find the link between the bit pattern of the encoded messages arriving at the mix and the decoded messages departing from the mix. The usage of the word encoded and decoded instead of encrypted and decrypted serves to highlight that the former operations are only used to achieve unlinkability, and not confidentiality, as may be understood to be the aim of encryption. Indeed, modifying RSA or any other encryption and decryption functions to provide unlinkability against passive or active attackers is a problem studied in depth in the context of the design of Mixminion.

Pfitzmann and Pfitzmann show in [161] that Chaum's scheme does not provide the necessary unlinkability properties. The RSA mathematical structure can be subject to active attacks that leak enough information during decryption to link ciphertexts with their respective plaintexts. Further *tagging attacks* are possible, since the encrypted blocks, using RSA are not in any way dependent on each other, and blocks can be duplicated or simply substituted by known ciphertexts. The output message would then contain two blocks with the same plaintext or a block with a known plaintext, respectively. Once again, the use of RSA in the context of a hybrid cryptosystem, in which only the keys are encrypted using the public key operations, and the body of the message using

a symmetric cipher were not very well studied at the time.

A further weakness of Chaum's scheme is its direct application of RSA decryption, which is also used as a signature primitive. An active attacker could substitute a block to be signed in the message and obtain a signature on it. Even if the signature has to have a special form, such as padding, that could be detected, a blinding technique could be used to hide this structure from the mix. It would be unfair to blame this shortcoming on Chaum, since he himself invented RSA blind signatures only a few years later [28].

The second function of a mix is to actually mix together many messages, to make it difficult for an adversary to follow messages through it, on a first-in, first-out basis. Therefore a mix batches a certain number of messages together, decodes them as a batch, reorders them in lexicographic order and then sends them all out. Conceptually, while bitwise unlinkability makes sure that the contents of the messages do not allow them to be traced, mixing makes sure that the output order of the messages does not leak any linking information.

In order to make the task of the attacker even more difficult, dummy messages are proposed. Dummy messages are generated either by the original senders of messages or by mixes themselves. As far as the attacker is concerned, they are indistinguishable in length and content from normal messages, which increases the difficulty in tracing the genuine messages. We will call the actual mixing strategy, namely the batching and the number of dummy messages included in the inputs or outputs, the *dynamic aspects* of mixing.

Chaum notes that relying on just one mix would not be resilient against subverted nodes, so the function of mixing should distributed. Many mixes can be chained to make sure that even if just one of them remains honest some anonymity would be provided. The first way proposed to chain mixes is the *cascade*. Each message goes through all the mixes in the network, in a specific order. The second way proposed to chain mixes is by arranging them in a fully connected *network*, and allowing users to pick arbitrary routes through the network. Berthold, Pfitzmann, and Standtke argue in [17] that mix networks do not offer some properties that cascades offer. They illustrate a number of attacks to show that if only one mix is honest in the network, the anonymity of the messages going through it can be compromised. These attacks rely on compromised mixes that exploit the knowledge of their position in the chain; or multiple messages using the same sequence of mixes through the network. Dingledine et al. argue in [68] that it is not the cascade topology that provides the advantages cited in [17]. Rather it is that messages proceed through the network in synchronous batches. They show that a synchronous free-route network provides better anonymity, against both passive and active attacks, than does a comparable cascade network. They also show that the network is more resilient to DoS in terms of both message delivery and the effects of DoS on anonymity.

Along with the ability for a sender to send messages anonymously to a receiver, Chaum presents a scheme by which one can receive messages anonymously. A user that wishes to receive anonymous email constructs an *anonymous return address*, using the same encoding as the header of the normal

messages. She creates blocks containing a path back to herself, and recursively encrypts the blocks using the keys of the intermediate mixes. The user can then include a return address in the body of a message sent anonymously. The receiver simply includes the return address as the header of his own message and sends it through the network. The message is routed through the network as if it was a normal message.

The reply scheme proposed has an important feature. It makes replies in the network indistinguishable from normal messages. In order to securely achieve this, it is important that both the encoding and the decoding operation provide bitwise unlinkability between inputs and outputs. This is necessary, because replies are in fact encoded when processed by the mix. The resulting message, after it has been processed by all the mixes in the chain specified by the return address, is then decoded with the keys distributed to the mixes in the chain. Both the requirement for decryption to be as secure as encryption, and for the final mix to know the decryption keys to recover the message, means that raw RSA cannot be used. Therefore, a hybrid scheme is proposed that simply encrypts a symmetric key in the header along with the address of the next mix in the chain. The symmetric key can be used to encrypt or decrypt the message. Since the keys are encoded in the return address by the user, they can be remembered by the creator of the reply block and used to decrypt the messages that are routed using them. Return addresses were also discussed in the Babel system [99] and implemented in the cypherpunk type I remailers. Unfortunately, other deployed systems like Mixmaster did not support them at all.

Chaum's suggestion that a receipt system should be in place to make sure that each mix processes messages correctly, has become a branch of anonymity research in itself, namely mix systems with verifiable properties. We will give an overview of these systems in section 4. A system was also proposed to support pseudonymous identities that was partly implemented as the Nym Server described in section 2.3.

3.2 ISDN mixes, Real Time mixes and Web mixes

In [158], Pfitzmann et al. designed a system to anonymize ISDN telephone conversations. This design could be considered practical, from an engineering point of view, since it met the requirements and constraints of the ISDN network. Later the design was generalized to provide a framework for real-time, low-latency, mixed communications in [114]. Finally, many of the design ideas from both ISDN and Real Time mixes were adapted for anonymous web browsing and called Web Mixes [15]. Part of the design has been implemented as a web anonymizing proxy, JAP[7]. All three designs were the product of what could be informally called the Dresden anonymity community (although early research started in Karlsruhe). We will present the main ideas on which these systems are based together to facilitate understanding.

[7] `http://anon.inf.tu-dresden.de/`

A major trend in all three of the just-cited papers is the intent to secure anonymous communication, even in the presence of a very powerful adversary. It is assumed that this adversary would be able to observe all communications on the network (subsumes the so-called "global passive adversary"), modify the communications on the links by delaying, injecting or deleting messages, and control all but one of the mixes. While other designs, such as Mixmaster and Babel (that will be presented next), opted for a free route network topology, ISDN, Real Time and Web mixes always use cascades of mixes, making sure that each message is processed by all mixes in the same order. This removes the need for routing information to be passed along with the messages, and also protects the system from a whole set of intersection attacks presented in [17]. The debate between the pros and cons of cascade topologies has continued throughout the years, with debates (such as [53]) as well as work exploring the advantages of different topologies [38, 68].

The designs try never to compromise on security, and attempt to be efficient. For this reason, they make use of techniques that provide bitwise unlinkability with very small bandwidth overheads and few asymmetric cryptographic operations. *Hybrid encryption with minimal length* encrypts the header, and as much as possible of the plaintext in the asymmetrically encrypted part of the message. A stream cipher is then used to encrypt the rest of the message. This must be performed for each mix that relays the message.

Furthermore, it is understood that some protection has to be provided against active tagging attacks on the asymmetrically encrypted header. A block cipher with a globally known key is used to transform the plaintext before any encryption operation. This technique allows the hybrid encryption of long messages with very little overhead. It is interesting to notice that while the header is protected against tagging attacks, by using a known random permutation, there is no discussion about protecting the rest of the message encrypted using the stream cipher. Attacks in depth could be used, by which a partially known part of the message is XORed with some known text, in order to tag the message in a way that is recognizable when the message is decrypted. As we will see Mixmaster protects against this using a hash, while Mixminion makes sure that if modified, the tagged decoded message will contain no useful information for the attacker.

From the point of view of the dynamic aspects of mixing, ISDN, Real Time and Web mixes also introduce some novel techniques. First the route setup messages are separated from the actual data traveling in the network. In ISDN mixes, the signaling channel is used to transmit the onion encoded message that contains the session keys for each intermediary mix. Each mix then recognizes the messages belonging to the same stream, and uses the session key to prime the stream cipher and decode the messages. It is important to stress that that both "data" messages and "route setup" messages are mixed with other similar messages. It was recognized that all observable aspects of the system such as route setup and end, have to be mixed.

In order to provide anonymity for both the initiator and the receiver of a call, rendezvous points were defined. An initiator could use an anonymous label

attached to an ISDN switch in order to be anonymously connected with the actual receiver. This service is perhaps the circuit equivalent of a Nym server that can be used by message-based systems. It was also recognized that special cases, such as connection establishment, disconnection and busy lines could be used by an active attacker to gain information about the communicating party. Therefore a scheme of *time slice* channels was established to synchronize such events, making them unobservable to an adversary. Call establishment, as well as call ending have to happen at particular times, and are mixed with, hopefully many, other such events. In order to create the illusion that such events happen at particular times, real or cover traffic should be sent by the users' phones through the cascade for the full duration of the time slice. An even more expensive scheme requires users to send cover traffic through the cascade back to themselves all the time. This would make call initiation, call tear-down and even the line status unobservable. While it might be possible to justify such a scheme for ISDN networks where the lines between the local exchange and the users are not shared with any other parties, it is a very expensive strategy to implement over the Internet in the case of Web mixes.

Overall, the importance of this body of work is the careful extension of mixes to a setting of high-volume streams of data. The extension was done with careful consideration for preserving the security features in the original idea, such as the unlinkability of inputs and outputs and mixing all the relevant information. Unfortunately, while the ideas are practical in the context of telecommunication networks, where the mix network is intimately linked with the infrastructure, they are less so for widely deployed modern IP networks. The idea that constant traffic can be present on the lines so that anonymity can be guaranteed, but can still be relatively low, is not practical in such contexts. Onion routing, presented in Section 5, provides a more flexible approach that can be used as an overlay network, but it is at the same time open to more attacks in principle if not in practice. The techniques we have just presented may nonetheless become increasingly relevant if fixed rate traffic, such as streaming data and VoIP, require strong anonymization.

3.3 Babel and Mixmaster

Babel [99] and Mixmaster [142] were designed in the mid-nineties, and the latter has become the most widely deployed remailer. They both follow a message-based approach, namely they support sending single messages, usually email, though a fully connected mix network.

Babel offers sender anonymity, called the "forward path" and receiver anonymity, through replies traveling over the "return path". The forward part is constructed by the sender of an anonymous message by wrapping a message in layers of encryption. The message can also include a return address to be used to route the replies. The system supports bidirectional anonymity by allowing messages to use a forward path, to protect the anonymity of the sender, and for the second half of the journey they are routed by the return address so as to hide the identity of the receiver.

While the security of the forward path is as good as in the (secured) original mix network proposals, the security of the return path is slightly weaker. The integrity of the message cannot be protected, thereby allowing tagging attacks, since no information in the reply address, which is effectively the only information available to intermediate nodes, can contain the hash of the message body. The reason for this is that the message is only known to the person replying using the return address. This dichotomy will guide the design of Mixminion, since not protecting the integrity of the message could open a system to trivial tagging attacks. Babel reply addresses and messages can also be used more than once, while messages in the forward path contain a unique identifier and a time-stamp that makes detecting and discarding duplicate messages efficient.

Babel also proposes a system of intermix detours. Messages to be mixed could be "repackaged" by intermediary mixes, and sent along a random route through the network. It is worth observing that even the sender of the messages, who knows all the symmetric encryption keys used to encode and decode the message, cannot recognize it in the network when this is done.

Mixmaster has been an evolving system since 1995. It is the most widely deployed and used remailer system. Mixmaster supports only sender anonymity, or in the terminology used by Babel, only anonymizes the forward path. Messages are made bitwise unlinkable by hybrid RSA and EDE 3DES encryption, while the message size is kept constant by appending random noise at the end of the message. In version two, the integrity of the RSA-encrypted header is protected by a hash, making tagging attacks on the header impossible. In version three, the noise to be appended is generated using a secret shared between the remailer, and the sender of the message, included in the header. Since the noise is predictable to the sender, it is possible to include in the header a hash of the whole message therefore protecting the integrity of the header and body of the message. This trick makes replies impossible to construct: since the body of the message would not be known to the creator of the anonymous address block, it is not possible to compute in the hash.

Beyond the security features, Mixmaster provides quite a few usability features. It allows large messages to be divided in smaller chunks and sent independently through the network. If all the parts end up at a common mix, then reconstruction happens transparently in the network. So large emails can be sent to users without requiring special software. Recognizing that building robust remailer networks could be difficult (and indeed the first versions of the Mixmaster server software were notoriously unreliable) it also allowed messages to be sent multiple times, using different paths. It is worth noting that no analysis of the impact of these features on anonymity has ever been performed.

Mixmaster also realizes that reputation attacks, by users abusing the remailer network, could discredit the system. For this reason messages are clearly labeled as coming from a remailer and black lists are kept up-to-date with email addresses that do not wish to receive anonymous email. While not filtering out any content, for example not preventing death threats being transmitted, at least these mechanisms are useful to make the network less attractive to email spammers.

3.4 Mixminion: the Type III Remailer

Mixminion [46] is the state of the art anonymous remailer. It allows for a fixed size message, of about 28 kbytes, to be anonymously transported over a set of remailers, with high latency. Mixminion supports sender anonymity, receiver anonymity via single-use reply blocks (SURBs), and bi-directional anonymity by composing the two mechanisms. This is achieved by mixing the message through a string of intermediate Mixminion remailers. These intermediate remailers do not know their position on the path of the message, or the total length of the path (avoiding partitioning attacks as described in [17]). Intermediate remailers cannot distinguish between messages that benefit from sender anonymity and anonymous replies.

Mixminion's first key contribution concerns the cryptographic packet format. Transported messages are divided into two main headers and a body. Each main header is further divided into sub-headers encrypted under the public keys of intermediary mixes. The main objective of the cryptographic transforms is to protect messages from tagging [160, 161]: an active adversary or corrupt node may modify a message, in the hope that they will be able to detect the modification after the message has been mixed. This would allow an adversary to trace the message and compromise anonymity. Mixmaster solves this problem by including an integrity check in the header read by each intermediary: if tampering is detected the message is dropped at the fist honest mix. Mixminion cannot use a similar mechanism, because of the need to support indistinguishable routing of anonymous replies. Instead, it relies on an all-or-nothing encryption of the second header and the body of the message, which is very fragile. Tampering cryptographically results in the address of the final receiver and the message being destroyed. The cryptographic format was designed to be well understood, and as a result it is quite conservative and inefficient.

The Minx packet format aims to provide the same properties as Mixminion at a lower computational cost and overhead [47]. It relies on a single pass of encryption in IGE mode, that propagates ciphertext errors forward. As a result, modifying the message results again in all information about the final receiver and the message being destroyed. Since all messages look random, no partial information is ever leaked through tampering. The original design of Minx provided security arguments but not a security proof. Shimshock et al. later showed that the original design could not be proven secure, but they also provided a small modification to Minx and proved that the resulting format and protocol is secure [177].

Mixminion uses a TCP based transport mechanism, that can accommodate link padding. Messages are transferred between remailers using a TLS protected tunnel, with an Ephemeral Diffie-Hellman based key exchange to provide forward security. This renders any material gathered by a passive adversary useless, since it cannot be presented to a remailer operator for decryption after the ephemeral keys are deleted. It also detects active adversaries that try to corrupt data traveling on the network. Therefore an adversary must be running malicious nodes to attack the network.

Two proposals have been put forward to strengthen the forward security and compulsion resistance of Mixminion mixing. The first, in [37], assumes that any communication leaves a trail of keys on intermediate mixes that can be used to decrypt future communications. Once a key is used, it is deleted or updated using a one-way function. Since subsequent messages may be dependent on previous messages for decoding, a mix that honestly deletes keys cannot decrypt intercepted messages upon request. Furthermore, an adversary needs to intercept and request the decryption of many messages in order to retrieve the key material necessary to decode any particular target message. The second technique [43] relies on the fact that the genuine receiver of an anonymous reply can pretend to be a relay, and pass the message to another pre-determined node. This assumes a peer-to-peer remailer system, and may be an incentive to run a Mixminion server.

The implementation of Mixminion brought to the surface many practical questions. Since the transport of Mixminion messages is unreliable, it is important to implement mechanisms for retransmissions and forward error correction. Such mechanisms are not trivial to implement and may lead to traffic analysis attacks. In order to be secure, all clients must know the full network of remailers. This has proved to be a scalability problem, and a distributed directory service had to be specified in order to distribute this information. Robust mechanisms for vetting nodes, and ensuring their honest behavior are still elusive. Practical deployment and integration into familiar clients has also been a challenge.

3.5 Foiling blending attacks

As we saw above, Babel and Mixmaster implement a traditional mix network model. They also both extend the original idea of mixing batches of messages together to feeding back messages in a pool, in the case of Mixmaster, or to delaying a fraction of messages an additional round, in the case of Babel. Such mix strategies, along with others, are designed to resist an $(n-1)$ *attack*, in which the adversary sends one message to be traced to an empty mix, together with adversary-recognizable messages. When the mix flushes, the only message that cannot be recognized is the one to be traced, which compromises anonymity.

In [173], Serjantov, Dingledine, and Syverson explain the attack as occurring in two phases: a flood and a trickle. In the first phase the adversary tries to flush all genuine messages from the mix by flooding while blocking any more of these from entering. Next, he trickles in the target message(s) along with other adversary messages. The $(n-1)$ attack is described as a specific instance of a more general kind of attack they called a *blending attack* since the number of adversary messages and honest messages to be blended together may be manipulated but the number of target messages might be other than 1, by intention or because the available control is not precise enough. It may also be that the number of other honest messages in the mix cannot be reduced to zero with certainty, so effectively the number of non-target messages is other than n. These numbers could be known but not exactly controlled or they could be unknown and only a distribution over them available. Serjantov et al. set out a

taxonomy of blending attacks based on what the adversary can determine and what resources must be used for different mixing strategies. In [148], O'Connor gives a rigorous analysis of the susceptibility of many different mix strategies to blending attacks.

A simple strategy proposed to counter such attacks is admission control, through authentication and ticketing systems [16]. If each user is properly authenticated when sending a message, flooding can be detected and foiled. This solution is not fully satisfactory though, since corrupt mixes may also inject messages. Having to authenticate may also reduce the perception of security offered by the system.

In [56], Diaz and Serjantov introduced a model for representing the mixing strategies of pool mixes. This model allows for easy computation of the anonymity provided by the mixing strategy towards active and passive adversaries. It was noted that $(n-1)$ attacks on pool mixes were favored by the deterministic dependency of the number of messages forwarded in any round and the number of messages kept in the pool for future rounds. The adversary could use this knowledge to optimize his efforts in terms of time and number of messages generated and have 100% certainty on the detection of the target at the output of the mix. In order to increase the effort and the uncertainty of the attacker, they propose randomizing the number of messages forwarded, as a binomial distribution of the number of messages contained in the pool. The randomization can be done almost for free: at the time of forwarding: instead of choosing a fixed number of random messages from the pool, the mix simply flips a biased coin for each message.

The first effect of the randomization is that the attacker succeeds only probabilistically, and the effort of the attacker increases as he tries to increase his probability of success. In [54] Diaz and Preneel analyze the robustness of various combinations of mixing and dummy generation strategies to $(n-1)$ attacks. They show that the combination of binomial mixing and randomized dummy generation strategies sets a lower bound on the anonymity of the target message. The adversary is able to significantly reduce the anonymity set of the message but he does not uniquely identify the message at the output of the mix. The protection offered to the message is proportional to the amount of dummies generated by the mix. Detailed analysis of $(n-1)$ and other blending attacks as well as the results and costs of deploying them are presented in [52, 54, 56, 171, 173].

Stop-and-Go mixes [121] (sg-mixes) present a mixing strategy, that is not based on batches but delays. They aim at minimizing the potential for $(n-1)$ attacks. Each packet to be processed by an sg-mix contains a delay and a time window. The delay is chosen according to an exponential distribution by the original sender, and the time windows can be calculated given all the delays. Each sg-mix receiving a message, checks that it has been received within the time window, delays the message for the specified amount of time, and then forwards it to the next mix or final recipient. If the message was received outside the specified time window it is discarded. This security feature was, however, not implemented in the practical implementation of sg-mixes, called '*Reliable*', which inter-operated with the pool mixes of the Mixmaster network. A practical

comparison on the anonymity provided by both the pool and sg nodes of the Mixmaster network towards passive adversaries is presented in [55]. This paper shows that, even in very low traffic conditions, the pool nodes provide a large anonymity set for the messages they route at the expense of longer delays. The Reliable node, which does not adapt the delay to the traffic load, provides no anonymity in extreme cases.

A very important feature of sg-mixes is the mathematical analysis of the anonymity they provide. Assuming that the messages arriving to the mix follow a Poisson distribution, it is observed that each mix can be modeled as a $M/M/\infty$ queue, and a number of messages waiting inside it follow the Poisson distribution. The delays can therefore be adjusted to provide the necessary anonymity set size.

The time window is used in order to detect and prevent $(n-1)$ attacks. It is observed that an adversary needs to flush the sg-mix of all messages, then let the message to be traced through and observe where it goes. This requires the attacker to hold the target message for a certain time, necessary for the mix to send out all the messages it contains and become empty. The average time that the message needs to be delayed can be estimated, and the appropriate time window can be specified to make such a delayed message be rejected by the mix.

Alpha mixing and Tau mixing are generalizations of a sort to stop-and-go mixing, introduced by Dingledine, Serjantov, and Syverson in [67]. The initial goal of these forms of mixing was to permit a mix to blend together traffic with different security sensitivities. The sender can potentially choose the desired tradeoff of latency and security, but all traffic still provides mutual protection benefit despite the diversity. The basic idea of an alpha level is that it determines the number of mix batches that are processed from when a message arrives until it is sent, a tau level uses the number of messages that must be processed rather than batches. These can be combined with time-delay requirements or pool strategies as well.

A different solution to blending attacks, the rgb-mix [48], is based on a controlled level of cover traffic. In this scheme by Danezis and Sassaman, each mix in the network sends 'red' heartbeat messages back to itself through the mix network. If at some point such messages stop arriving it may mean that the mix is subject to the first phase of a blinding attack. The mix then responds by injecting 'green' cover traffic to confuse the adversary. The key property that makes this scheme secure is the inability of the adversary to tell apart genuine messages, to be blocked, and heartbeat messages that need to be let through for the mix not to introduce additional cover traffic. Under normal operating conditions the traffic overhead of this scheme is minimal, since additional traffic is only introduced as a response to attack.

4 Robust and verifiable mix constructions

Chaum's original mix network design included a system of signed receipts to assure senders that their messages have been properly processed by the network.

A whole body of research was inspired by this property and has attempted to create mix systems which are robust against subverted servers denying service, and that could offer a proof of their correct functioning alongside the mixing. Such systems have been closely associated with voting, where both universal verifiability of vote delivery and privacy are of great importance.

Most of the proposed schemes use mix cascades. For this reason, no information is usually communicated between the sender of a message and intermediate mixes in the cascade. It is assumed that routing information is not necessary, since mixes process messages in a fixed order. The first scheme to take advantage of this was the *efficient anonymous channel and all/nothing election scheme* proposed by Park, Itoh, and Kurosawa in [156]. In this system, a message is an ElGamal ciphertext of fixed length (length is independent of the number of mixes it goes through). Furthermore, the scheme uses a *cut and choose* strategy, which makes it all-or-nothing, meaning that if any of the ciphertexts is removed from a mix batch, then no result at all is output. This property assures that partial results do not affect a re-election.

Chaum's original design was for decryption mixes, in which encryption for each mix in a route is layered on the message by the sender and then stripped off by the relevant mix as it processes that message. Besides assuming a cascade topology, many of the designers of robust or verifiable mix systems follow Park et al. by instead using a re-encryption approach. In this approach the message is typically public-key encrypted just once using a homomorphic encryption scheme such as ElGamal. Each mix then re-encrypts the message with a new randomization values so that it changes its appearance, which is why the messages can be of fixed length, indepent of path length. All messages sent into the cascade are encrypted using the same public key. For applications like sending of anonymous email to arbitrary recipients at various destinations this would not be very useful. For applications such as municipal elections, however, it fits quite naturally.

Birgit Pfitzmann found two attacks against the Park, Itoh, and Kurosawa proposal [160]. The first attack is very similar to an earlier one against a different mix design that we discussed above [161] and makes use of characteristics that are invariant at the different stages of mixing because of the ElGamal cryptosystem. She also found an active attack, wherein the input ElGamal ciphertext is blinded by being raised to a power, which results in the final output also being raised to this power. This is a chosen ciphertext attack againsth which a lot of systems struggle, and eventually fail. Pfitzmann also notes that the threat model assumed is weaker than the one proposed by Chaum. A dishonest sender is capable of disrupting the whole network, which is worse than a single mix, as is the case in Chaum's paper. She does not propose any practical countermeasures to these attacks: any straightforward fix would compromise some of the interesting features of the system.

In parallel with Birgit Pfitzmann's work, Kilian and Sako proposed a *receipt-free mix-type voting scheme* [122]. They attempt to add universal verifiability to [156], which means that all senders will be able to verify that all votes were taken into account, not simply their own. They also highlight that many verifi-

able mix schemes provide, at the end of mixing, a receipt that could be used to sell or coerce one's vote. They thus attempt to make their system *receipt-free*. They do this by forcing each mix to commit to its inputs and outputs, and prove in zero knowledge that it performed the decryption and shuffle correctly. Unfortunately, Michels and Horster show in [138] that the scheme is not receipt-free if a sender collaborates with a mix, and that the active attacks based on blinding proposed by Birgit Pfitzmann could be used to link inputs to outputs.

In order to avoid disruption of the system if a subset of mixes is subverted, Ogata et al. propose a *fault tolerant anonymous channel* [149]. This uses a threshold cryptosystem to make sure that a majority of mixes can decode messages even if a minority of mixes do not participate. Two systems are proposed, one based on ElGamal and the other based on the r^{th} residue problem. A zero knowledge proof of correct shuffling is also proposed for the r^{th} residue problem.

In 1998, Abe presented a mix system that provided universal verifiability and was efficient, in the sense that the verification work was independent from the number of mix servers [1]. Abe also shows an attack on [122], that uses the side information output for the verification to break the privacy of the system. He then presents a mix system that works in two phases, ElGamal re-encryption and then threshold decryption. The first phase is proved to be correct before the second can proceed, and then a proof of correct decryption is output at the end of the second stage.

The systems that provide universal verifiability based on proofs of permutations and on zero-knowledge proofs are computationally very expensive. In [110], Jakobsson designs the *practical mix* to reduce the number of expensive operations. In order to prove the correctness of the shuffle, novel techniques called *repetition robustness* and *blinded destructive robustness* are introduced. The network works in two phases: first, the ciphertexts are ElGamal blinded, and then, the list of inputs is replicated. Each of the replicated lists is decoded by all mixes, which results in lists of blinded plaintexts. The resulting lists are sorted and compared. If all elements are present in all lists, then no mix has tampered with the system and the unblinding and further mixing can proceed. Otherwise, a sub-protocol for cheater detection is run. While being very efficient, the practical mix has not proved to be very secure, as shown by Desmedt and Kurosawa [51]. They show that one subverted mix in the practical mix can change ciphertexts and still not be detected. They then introduce a new mix design, in which verification is performed by subsets of mixes. The subsets are generated in such a way that at least one is guaranteed not to contain any subverted mixes.

In an attempt to further reduce the cost of mixing, Jakobsson introduced the *flash mix*, that uses re-encryption instead of blinding to keep the number of exponentiations down [111]. As in the practical mix, mixing operates in many phases, and uses *repetition robustness* to detect tampering. Furthermore, two dummy messages are included in the input. These are de-anonymized after all mixes have committed to their outputs in order to make sure that attacks such as in [51] do not work. Mitomo and Kurosawa found an attack against flash mixing nonetheless, fixed by changing the unblinding protocol [139].

A breakthrough occurred when Furukawa and Sako [83] and Neff [146] proposed efficient general techniques to universally verify the correctness of a shuffle of ElGamal ciphertexts. The first of these provides proof that the matrix used was a permutation matrix, and the second uses verifiable secret exponent multiplication to improve its efficiency.

Even though the above techniques are more efficient than any other previously known, they are still not efficient enough to scale for elections with millions of participants. For this reason, Golle et al. [95] proposed optimistic mixing, mixing that works quickly if there is no attack detected, but provides no result if an error occurs. In the error case, it provides a fall back mechanism for a more robust technique (such as in [146]) to be used. Each mix in the chain outputs a "proof" of permutation, that could be faked by tampering with the ciphertexts. This is detected by making the encryption plaintext-aware. The second decryption, revealing the votes, is only performed if all outputs of mixing are well-formed. Douglas Wikström found a series of attacks against this scheme [195, 197]. The first two attacks are closely related to attacks in [160], mentioned above, and can break the anonymity of any user. Another attack is related to [51] and can break the anonymity of all users and compromise robustness. Finally, attacks based on improperly checking the ElGamal elements are also applicable, and are further explored in [196].

A serious drawback of traditional robust mix cascades is that each mix has to wait for the output of the previous mix before processing messages. This means that the latency increases with the number of mixes, and that most of the time mixes perform no computations. In [93], Golle and Juels present a technique that allows for universally verifiable parallel mixing in four steps of communication and the equivalent of two steps of mixing. Their techniques drastically reduce the latency of mixing, but Borisov shows that when multiple input messages are known to the adversary, the anonymity provided by this technique is far from optimal [21].

A fundamentally new way of looking at robust mixing is presented in [3]: mixing is seen as a computation to be outsourced to a third party. Yet, this third party should gain no information about the actual shuffle. Two protocols that implement such an algorithm are presented, based on Paillier and BGN homomorphic encryption. The third party accepts a set of ciphertexts, and participates in an obfuscated mixing algorithm to produce a re-encrypted and shuffled set of outputs. Despite only public keys being used, neither the third party, nor any observer, can link inputs and outputs.

4.1 Universal Re-encryption and RFID Privacy

A hopeful line of research looks at extending robust cascades into general mix networks. These may use nondeterministic decoding and routing protocols, possibly implemented using the new universal re-encryption primitive introduced by Golle, Jakobsson, Juels, and Syverson in [91], and improved for space efficiency by Fairbrother in [75]. Universal re-encryption was introduced to reduce trust in individual mixes by not requiring them to protect or even know any keys: The

public key needed to re-encrypt (transform) the message when passing through a mix is hidden within the message itself. This also means that, unlike ordinary re-encryption mixes, messages encrypted under different keys can be processed together in the same batch. The same mathematical properties that make this possible make it malleable. So, arbitrary message submissions to a mix cannot be permitted. The designs of many systems that use it [96, 124, 125, 132, 133] have not been adequately cognizant of its malleability, and they were found to be vulnerable to tagging attacks in [41]. The original universal re-encryption scheme in [91] avoided such problems by requiring proofs of knowledge of the plaintext and encryption factors for any encryped message that is submitted in order to be deemed secure.

In [91], Golle et al. also described potential application of universal re-encryption to RFID tags. RFID tags are now used for a variety of commercial applications. These include automated payment for highway tolls, mass-transit, gasoline purchase, etc.; inventory tracking and control; shelving, circulation, and inventory of library books; identification of recovered lost or runaway pets; checking of passports and identification documents; and many other uses. Primary discussion of RFID tags is covered in another chapter of this book.

Numerous security and privacy concerns have been raised about the use RFID tags, along with many proposals to address these on both a technical and a policy level. Using universal re-encryption, mixing tag readers could rewrite tags that might have been encrypted under multiple different public keys. This would make it more difficult to track individuals carrying multiple RFID tags, e.g., tags on goods purchased while wandering through various shops. However, the authors noted that there is a potential danger of tracking if ciphertexts are not properly formed, so that mixing readers should check for this. No suggestion was offered, however, of how to prevent tracking an individual by an adversary who wrote to a tag the individual was carrying using the adversary's own public key. This was answered by Ateniese, Camenisch and de Medeiros when they introduced, insubvertible encryption, a variant of universal re-encryption, in [7]. Insubvertible encryption allows RFID tags to be initialized with certificates, and only encryptions that match the certificate are mixed and written to a tag by honest mixing tag readers. Illegitimate ciphertexts can be replaced with random untraceable text, which will not be readable by the owner of the tag but will not be traceable either.

4.2 Provable security for mix-networks

While most designs for robust mix nets use pure ElGamal encryption, some provide solutions for hybrid schemes. In [150], Ohkubo and Abe present a hybrid mix without ciphertext expansion. Jakobsson and Juels also present a scheme that is resistant to any minority coalition of servers [112]. Möller presents the security properties of a mix packet format in [141]. Camenisch and Lysyanskaya presented further work in proving packet formats correct in [26]. Other packet formats attempt to provide specialized properties: Golle uses packet format to allow a mix to prove that a particular output was an input to the mix, clearing

the operator from any suspicion that they injected the message [90].

Reputation based schemes have also been used to increase the reliability of mix networks in [61], and mix cascades in [69]. Both these papers show ways to replace the *statistics pages* compiled in the Mixmaster system using *pingers* [154] with a more robust system to determine which nodes are reliable and which are not. Users can then choose reliable nodes, or the system can exclude unreliable ones from directories. The idea is not to provide provable guarantees of processing but only to increase the likelyhood that reliable mixes are chosen for processing. Statistical guarantees are given that mixes are processing messages properly, and the overhead and cryptographic complexity of these schemes are lower than for mix systems with shuffle proofs, etc. They are thus probably better suited to general communication applications than to those where mishandling of even single messages must be prevented and/or detected with very high probability. Like the later rgb mixes mentioned above, these use test messages to monitor the network. In principle these can provide arbitrarily high probability of detecting mishandling (below certainty), assuming that the overhead of the test messages is not too high to be practical or to remain indistinguishable from other messages.

Jakobsson, Juels, and Rivest take another probabilistic approach to detecting mishandling via *randomized partial checking*, presented in [113]. Randomized partial checking allows universal verifiability to be implemented on generic mix networks. In this scheme, all mixes commit to their inputs and outputs and then are required to disclose half of all correspondences. This assures that if a mix is dropping messages, it will be quickly detected. Privacy is maintained by pairing mixes and making sure that the message is still going through enough secret permutations. For safety, it was proposed that mixes be paired, and when one in a pair is required to disclose a correspondence the other is required to keep the relevant correspondence secret in order to ensure that enough mixing is performed for each message. In [97] it is shown, using path coupling tools and graph theory, that such caution is not necessary, since messages will mix with high probability after $\log N$ steps, even if correspondences are revealed at random.

A separate line of research attempts to prove the mixing, and hence privacy, properties provided by the network instead of the robustness. Such systems are usually expensive, since they rely on extensive amounts of cover traffic to provably ensure that no information about the actual traffic patterns is leaked. Systems presented in [162] and [14] are in this tradition. The latter proves that in a random network of communications, one could embed a very large number of possible sub-networks of a certain butterfly-like form, and show that, at each step, messages are mixed together with high probability. Interestingly, Klonowski and Kutyłowski prove that traditional mix networks mix all input messages after $\log N$ rounds, despite the presence of adversaries that reveal to each other the path of messages [126].

Some mix strategies are designed to be fragile on purpose. If a single message gets deanonymized through compulsion, then all the messages get deanonymized [167], or, alternatively, a secret of the mix operator can easily

be inferred [94]. The latter provides operators with specific incentives to resist compulsion, while the incentive in the first scheme is in making it more difficult to trace a one message or a small number of messages discretely.

5 Onion routing

Onion routing is an anonymity approach designed to use circuits and designed to facilitate bidirectional, low-latency communication. Its security comes from obscuring the route of the circuit and its low-latency practicality comes in part from using computationally expensive cryptography (e.g., public-key) only to lay the circuit; data is passed over the circuit protected by less expensive cryptography (e.g., symmetric-key). Onion routing is thus somewhat like a free-route mix network, except that onion routers are not mixes. They do not obscure the relation between the order of data packets entering and leaving the onion router. Since the initial onion-routing design was introduced by NRL (U.S. Naval Research Laboratory) researchers in 1996 [89], there have been numerous onion-routing systems designed and deployed with various features. The common thread that makes them all onion routing is the building of circuits along obscured paths and the efficient choice of cryptography used so as to separate circuit building from data passing.

5.1 Early onion routing systems

In the initial design, instead of routing each anonymous packet separately, the first message opens a circuit through the network. The circuit is labeled with different identifiers at each onion router in the path. By matching inbound and outbound identifiers, onion routers are able to route each message along this predetermined path. Finally, a message can be sent to close the path. Often, we refer to the application level information traveling in each of these labeled circuits as an anonymous stream.

The first message sent through the network is encrypted in layers that can only be decrypted by a chain of onion routers using their respective private keys. This first message contains key material shared between the original sender and the onion routers, as well as the address of the next node. This message is the *onion* that gives the design its name—because the content of this message is effectively the layers themselves; it is composed of nothing but layers. As with Chaum's mixes, care is taken to provide bitwise unlinkability, so that the path that the first message takes is not trivial to follow just by observing the bit patterns of messages. There is also link encryption which reduces this purely to an insider threat. Loose routing is also proposed, according to which routers relay streams through paths that are not directly specified in the original path opening message. The hope was that such a scheme would permit circuits to continue being built even when an onion router does not have a direct connection to the next onion router specified in the onion.

Data traveling in an established circuit is also encrypted in layers, but using

the symmetric keys distributed to the onion routers. These are termed "data onions"; although they do carry content separate from the layers themselves. Labels are used to indicate the circuit to which each packet belongs. Different labels are used on different links, to ensure bitwise unlinkability, and the labels on the links are encrypted using a secret shared key between pairs of onion routers. All communication also passes through the network in fixed-size cells. This prevents a passive observer from using packet size to determine which packets belong to the same anonymous stream, but it does not hide this information from a subverted onion router.

The objective of onion routing is to make traffic analysis harder for an adversary. It aims first at protecting the unlinkability of two participants who know each other from third parties, and secondly, at protecting the identities of the two communicating parties from each other.

Onion routing admits to being susceptible to a range of attacks. It has always been expected, and has been confirmed by analysis that, in the absence of heavy amounts of cover traffic, patterns of traffic are present that could allow a passive attacker to match a stream where it enters and exits the network to identify the communicating parties. Such attacks have been called timing attacks and end-to-end correlation attacks. While they have been cited in the literature [164] for many years, and details of how they work and how effective they are have been presented [40, 129, 152, 174, 193, 200, 201].

To explore how much protection might be possible against timing attacks, the next generation of design from NRL added realtime mixing of data packets traveling through an onion router on different streams and link-level traffic shaping (padding and limiting) based on a sliding-window weighted average of prior traffic [165, 183]. Also added to both this design and the subsequent NRL design, Tor, were protocol data structures to permit partial route padding between the client and onion routers in the path to complicate traffic analysis by insiders on the the path.

Unlike ISDN mixes [158], even this mixing onion-routing design does not perform any synchronous mixing of requests for opening or closing channels. They only are mixed with whatever other streams are active on the same onion routers. It is very unlikely that the anonymity of circuit-setup messages can be maintained against participating onion routers unless circuits are set up synchronously, which was not part of the design. Indeed, Bauer et al. showed that very effective timing attacks against Tor were possible using only the circuit setup messages [11]. Therefore, an attacker could follow such messages and compromise the anonymity of the correspondents. Of course this is somewhat less of a concern in a free-route network than in a cascade, such as in ISDN mixes and Web mixes [15], where it is known exactly where to watch for corresponding channels opening and closing (which is why those mix designs do assume synchronous circuit setup). Furthermore, it is trivial for an active attacker on the circuit to induce a timing signature even if padding adequately obscures any passive signature. Besides timing, it was also understood even at the initial design that an attacker at any point in the path could corrupt messages being sent and have this be visible to the last onion router. This is similar

to the tagging attack against Mixminion described above. Because onion routing is circuit-based and low-latency, however, the countermeasures Mixminion employs are pointless and are not available. Given this, and as research failed to yield effective resistance against active attacks on low-latency systems, and given the network cost of the techniques that were proposed to counter even passive attacks, the third generation of onion-routing design from NRL, Tor, returned to the original approach of simply not doing any padding and just assuming vulnerability to timing correlation. The Tor design will be discussed below.

In order to make deployment easier, it was recognized that some users would be unwilling or unable to be servers. The second-generation design thus broke away from the pure peering of the first generation and allowed onion routing clients that built circuits but did not route traffic for others. Also, some onion routers might wish to only serve particular clients. The concept of *entrance policies* was added to onion routing to encapsulate this [165, 182], allowing routers to decide which network connections they were configured to serve. Onion routers are also free to peer with only a subset of other routers, with which they maintain longstanding connections. It was similarly assumed that some onion routers might wish to only allow traffic exiting the onion routing network through them to contact particular network locations and/or ports (protocols). Thus, *exit policies* were also added. These could be advertised so that routes could be built to appropriate network exit points. This separation of basic client from onion routing server and the permitted flexibility for entrance and exit policy were probably the most significant changes from the original design that would facilitate the ultimate widescale adoption of onion routing. Usability and flexibility for both individuals and server operators allows them to participate in the system in a way consistent with their diverse interests and incentives, which encourages the system to take off and grow in ways not otherwise likely [2, 63].

There were independently deployed onion routing networks in the late 1990s using the NRL designs and code, but the first independently designed onion-routing system was the Freedom Network, built and deployed from late 1999 to late 2001 by Zero Knowledge Systems, a Canadian company (now called Radialpoint and not offering anything like Freedom amongst its current products or services). The principal architect of the network was Ian Goldberg [86] who published with Adam Shostack and others a series of white papers describing the system at various levels of detail [8, 9, 23, 87, 88]. This system had many important differences from the NRL designs: UDP transport as opposed to TCP, block ciphers rather than stream ciphers used for encrypting the data, a significant integrated pseudonym and certificate system, etc., but the basic route building and message passing of the network followed essentially that of the first two generations of NRL design described above.

One of the most important contributions of Freedom was that it was a significant commercial deployment. The first-generation prototype that NRL ran from 1996 to 2000 processed connections from tens of thousands of IP addresses and averaged 50000 connections per day during its peak months, but these all ran through a small five node system at NRL. It was not actually distributed at all.

Some second-generation deployments had more than a dozen nodes scattered across the U.S. and Canada, but these were testbeds that saw neither large-scale nor public use. Amongst other things, Freedom was an onion-routing system with a widely distributed set of independently-run server nodes and twenty five thousand subscribers at its peak. This is significant as the first successful demonstration that a large-scale deployment and operation of an onion-routing system is technically feasible.

5.2 Tor's Onion Routing

In 2003, the NRL onion routing program began deployment of a third generation design called 'Tor'.[8] The initial Tor design was published in 2004 [65]. Tor relays arbitrary TCP streams over a network of relays, and is particularly well tuned to work for web traffic, with the help of the Privoxy[9] content sanitizer.

In Tor's network architecture a list of volunteer servers is downloaded from a directory service. Then, clients can create paths by choosing three random nodes, over which their communication is relayed. Instead of an 'onion' being sent to distribute the cryptographic material, Tor uses an iterative mechanism. The client connects to the first node, then it requests this node to connect to the next one. The bi-directional channel is used at each stage to perform an authenticated Diffie-Hellman key exchange. This guarantees forward secrecy and compulsion resistance: only short term encryption keys are ever needed. This mechanism was first described in 2002 as part of the onion-routing design, Cebolla [25]. This is different from both the first two generations of onion routing from NRL and from the Freedom design. Authenticating that the right onion router is establishing a session key with the client is still done using RSA. There have since been published proposals to build the circuits entirely using Diffie-Hellman for greater efficiency [117, 153], but nothing has been implemented at time of writing. While these appear promising, they require further analysis before they can be incorporated into Tor. Interestingly, during the design of second-generation NRL onion routing, a move to using Diffie-Hellman for circuit establishment was contemplated. This was, however, purely on grounds of computational efficiency and not for any additional security properties it would provide, such as forward secrecy.

As mentioned above, Tor does not attempt to offer security against even passive observers of a circuit. We have already mentioned the traffic analysis techniques that have been developed throughout the years to trace streams of continuous traffic traveling in a low latency network [40, 129, 152, 174, 193, 200, 201]. A separate but related thread of research has been developed in the intrusion detection community. That approach tries to uncover machines used as stepping stones for an intrusion [20, 194]. The approach is difficult to foil,

[8] This was both a recursive acronym coined by Roger Dingledine for 'Tor's Onion Routing' and his way of telling people at the time that he was working on *the* onion routing from NRL rather than some other version of onion routing. Note that the recognized spelling is 'Tor' not 'TOR'.

[9] `http://www.privoxy.org/`

unless the latency of the traffic is high, or a lot of cover traffic is injected—both of which are very expensive. Tor instead opts for getting security though being highly usable and cheap to operate [10, 63].

Tor builds on the flexibility of running a client separate from a router node and flexible entrance and exit policies introduced in the second-generation design. Since onion routing gets its security from obscuring routes but is vulnerable to timing attacks, security against an adversary corrupting or observing c nodes in a network of n onion routers is at best c^2/n^2 [79, 183]. The easier it is for independent people to run nodes in the network, the better the security of the system. By making the network volunteer based, Tor removed some of the difficulties that come with having a commercial deployment and also a scalable one [2]. Other usability innovations added to Tor include the introduction of a controller protocol separate from the basic Tor routing functionality and a GUI, Vidalia[10], that was originally created by Matt Edman and Justin Hipple. In addition to other features, the GUI provides a simple interface to configure running either a server or a client, and the choice to turn a client into a server can be enacted with a click. TorButton is also a simple Firefox plugin that, among other things, allows routing through Tor to be turned on or off at the touch of a button. Tor also now comes via a GUI installer that bundles Tor, Vidalia, Privoxy, and TorButton. Documentation for Tor and related programs has been translated into at least 17 languages at the time of writing. All of these contribute to making Tor more usable, hence more used and widely deployed, hence more secure [63, 66].

Its vulnerability against passive adversaries has made Tor fragile against previously unexplored attacks. First, a realistic assessment of the probability a single party can observe multiple points on the path is necessary. It turns out that the topology of the Internet is such that many, seemingly unrelated networks, are interconnected through hubs, or long distance links that can be observed cheaply by a single ISP, Autonomous System (AS), Internet Exchange (IX), or similar entity [73, 78, 144]. A second possible path for attack, presented by Murdoch and Danezis in [143], uses indirect network measurements to perform traffic analysis and does away with the assumption that a passive adversary needs local access to the communication to perform traffic analysis. An attacker relays traffic over all onion routers, and measures their latency: this latency is affected by the other streams transported over the onion router. Long term correlations between known signals injected by a malicious server and the measurements are possible. This allows an adversary to trace a connection up to the first onion router used to anonymize it. The published attack was done at a time when the network was orders of magnitude smaller than it is today (c. 35 nodes vs. thousands of nodes). It is an interesting and unknown question how effectively it would scale to the current network size. Recent work suggests that unchecked extension of path length can be leveraged to make such scaling feasible [74].

Tor also provides a mechanism for *hidden servers*. This is another instance

[10] http://www.vidalia-project.net/

of protecting the responder in anonymous communication, rather than just the initiator. Hidden services can be used to protect content from censorship and servers from DoS because it is hard to attack what cannot be found. A hidden server opens an anonymous connection and uses it to advertise a contact point. A client that wants to contact the server, goes to the contact point and negotiates a separate anonymous "rendezvous" channel used to relay the actual communication. An attack against this early architecture was demonstrated by Øverlier and Syverson in [152]. The intuition behind this attack is that an adversary can open multiple connections to the hidden server, sequentially or in parallel, and can control the flow of traffic towards the server on these. The adversary needs to control one corrupt router and to wait until his router is chosen by the server as the first node for the fresh anonymous path it builds to the rendezvous point. Then the adversary effectively controls two nodes on the anonymous path, one of which is next to the real server—-and the anonymity provided to the server is completely compromised. Though intersection attacks against anonymity systems were well known in the literature, this was the first such attack used in the wild. The speed with which this could be accomplished (typically several minutes) prompted the introduction of *guard nodes* for all Tor circuits. Since the time till end-to-end timing compromise of a circuit connecting repeated communicants was so short there was little value to spreading out the trust of the nodes contacted by the circuit initiator. Instead a small number of nodes is chosen as trusted and used the initial (guard) node for all circuits from that initiating client. Guard nodes are based on the more general idea of helper nodes for anonymity systems introduced by Wright et al. [199]. Øverlier and Syverson enhanced the efficiency of their attacks by misreporting of the bandwidth carried by the adversary node, causing it to be chosen with higher preference. It was speculated that with multiple nodes these attacks could easily be carried out against ordinary circuits, not just those built to connect to hidden services. This was later shown by Bauer et al. [11], who also showed that one could use such misrepresentation to enhance the chances of being chosen as a guard node. Countermeasures, such as limiting the self-reported bandwidth permissible, have since limited such attacks, and stronger countermeasures continue to be explored.

5.3 Peer-to-peer onion-routing networks

In Chaum's original work it is assumed that if each participant in the mix network also acts as a mix for others, this would improve the overall security of the network. Recent interest in peer-to-peer networking has influenced some researchers to further examine such networks with large, but transient, numbers of nodes. Although, as noted above, requiring all participants to run nodes does not necessarily encourage widescale adoption. Sometimes just the opposite is true with a concomitant effect on anonymity.

Tarzan [81] is a peer-to-peer network in which every node is an onion router. A peer initiating the transport of a stream through the network would create an encrypted tunnel to another node, and ask that node to connect the stream

to another peer. By repeating this process a few times, it is possible to have an onion-encrypted connection, relayed through a sequence of intermediate nodes.

An interesting feature of Tarzan is that the network topology is somewhat restricted. Each node maintains persistent connections with a small set of other nodes, forming a structure called *mimics*. Routes for anonymous messages are selected in such a way that they will go through mimics and between mimics in order to avoid links with insufficient traffic. A weakness of the mimics scheme is that the selection of neighboring nodes is done on the basis of a network identifier or address which, unfortunately, is easy to spoof in real-world networks.

The original Tarzan design only required each node to know a random subset of other nodes in the network [82]. This is clearly desirable due to the very dynamic nature of peer-to-peer networks and the volatility of nodes. On the other hand, Clayton and Danezis found an attack against this strategy in the early Tarzan design [42]. The attack relies on the fact that the network is very large, and nodes have a high churn rate. As a result any single node only knows a small subset of other nodes. An adversary node that is included on the anonymous path can tell that the originator of the connection knew three nodes: the corrupt node itself, its successor, and its predecessor. It turns out that those three nodes uniquely identify the originator of the stream with very high probability. This is very reminiscent of the attack described above that Shmatikov showed to be successful against Crowds [178]. The final version of Tarzan requires each node to know all others in order to fix this attack, which is clearly less practical. In addition to this epistemic fingerprinting of who knows which nodes in a network, it is also possible to perform epistemic bridging attacks based on who *does not* know particular nodes in the network [50].

MorphMix [168] shares a very similar architecture and threat model with Tarzan. A crucial difference is that the route through the network is not specified by the source but chosen by intermediate nodes, observed by witnesses specified and trusted by the user. While the attack by Danezis and Clayton does not apply to route selection, variants might apply to the choice of witness nodes.

The MorphMix design assumes that allowing the intermediate nodes to choose the route through the network might lead to *route capture*: the first subverted onion router on the path can choose only other subverted nodes to complete the route. For this reason, MorphMix includes a *collusion detection* mechanism that monitors for any cliques in the selection of nodes in the path. This prevents subverted nodes from routinely running attacks on the network but does not provide security in every case. In [186], Tabriz and Borisov presented an attack on the collusion resistance mechanism of MorphMix.

Another way to try building peer-to-peer anonymity networks is the structured approach [22, 32, 107, 145]. These systems generally build anonymous routes by looking up nodes to route through in a distributed hash table (DHT). By exploiting the structure implicit in the DHT and by adding randomness plus redundancy to the lookups, they make it difficult to either determine or direct the node discovery and selection done by the route builder. As with the above peer-to-peer anonymity systems, security for structured peer-to-peer anonymity systems has turned out to be quite elusive. In [140], Mittal and Borisov

studied AP3 [107] and Salsa [145] and found that both had significant leaks in the lookups associated with routing-node discovery and selection, leaks that allowed compromise of route-selection security by an adversary composed of a much smaller fraction of the total network than had previously been thought. In the case of Salsa, they showed an interesting tradeoff between mechanisms to prevent active attacks and passive observations of lookup. More redundancy helps resist active attacks, but leaks more information.

6 Other systems

A number of other anonymous communication systems have been proposed through the years. In [30], Chaum presents the dining cryptographers' network, a multi-party computation that allows a set of participants to have perfect (information-theoretic) anonymity. The scheme is very secure but impractical, since it requires a few broadcasts for each message sent and is easy to disrupt for dishonest users. A modification of the protocol guarantees availability against disrupting nodes [191]. Herbivore [85] uses DC-nets as part of a two-level anonymity system: users form small cliques that communicate within them using DC-nets. Finally, in [92] asymmetric techniques are described that make DC-nets robust against disruption.

Hordes uses multicast to achieve anonymity [130], while P5 uses broadcast-trees [176]. Buses use a metaphorical bus route that travels over all nodes carrying messages [12]. This is in fact a broadcast, and trade-offs between longer routes and more routes are discussed from an anonymity and latency perspective.

Traffic Analysis Prevention (TAP) systems attempt to provide third party anonymity, given a collaborating set of senders, receivers and relays. In [188], Timmerman describes adaptive traffic masking techniques, and a security model to achieve *traffic flow confidentiality* [187]. The information-theoretic approach to analyzing TAP systems is presented by Newman et al. in [147]. They study how much protection is offered overall to the traffic going through a TAP system, by creating a rigorous mathematical model of traffic analysis, rerouting and cover traffic. They also discuss how much anonymity can be achieved in this model given a budget that limits the total number of padding and message rerouting operations available to the network. This builds on their previous work in [190]. The research group at the Texas A&M University has a long-term interest in traffic analysis prevention of real time traffic [98]. Similarly, in [115] Jiang, Vaidya, and Zhao presents TAP techniques to protect wireless packet radio traffic.

7 Conclusions

Anonymous communications, despite being first proposed over 25 years ago, has become since 2000 an extremely active field of research. It is also increasingly

relevant since systems that are the direct result of this research, systems like Freedom, Tor, JAP and Mixminion, have been deployed and used to protect the privacy of thousands of people.

Anonymous communications research has also matured to the point that new systems must imperatively take into account the existing literature and ensure that they are not weak under known attacks and models. The aim of this chapter has been to present a road map of the most important systems-concepts and the key refinements to which they have been subject.

As in any mature field, new designs will inevitably have to mix and match from elements already present in older systems to best match their environment. Designs tailored to peer-to-peer systems or telephony are a prime example of this. Those systems are also a prime example of the care that a researcher must take when mixing and matching ideas: anonymous communications are fragile, and even simple modifications may lead to traffic analysis attacks.

A set of key observations must be in the minds of designers and researchers looking at anonymous communications in the future.

The *concepts of anonymity* in communication networks is a well understood problem. Definitions and metrics that express the anonymity properties of communications are available, and used to evaluate systems. Despite all security efforts, an upper limit on the anonymity that a system can provide is given by black box attacks: no matter how good the anonymity system is, effective attacks can be deployed in the long term by observing the edges of the anonymous communication network. As a result we say that the use of anonymous communication can be secured only tactically (for short periods) and not strategically or in the long term.

Concerning *trust models* the earliest anonymous communication systems relied on one central trusted server. This model has proven weak against compulsion, denial of service, traffic analysis carried out by a local eavesdropper, or maliciousness of the central node. Centralized trust models have been abandoned in favor of models where trust is distributed over a set of servers. Trusting a set of unknown, random nodes presents some serious limitations as well, particularly against attackers able to introduce a large number of corrupted nodes in the system (Sybil attacks [71]). The alternative is to trust nodes based on some knowledge about them or the links. As we have noted, researchers are just beginning to study the implications of routing based on trust [73, 116, 181].

Solutions for *high-latency applications* such as email have significantly evolved since the first schemes were proposed. The loose delay requirements allow for the design of secure solutions, providing a reasonably high resistance to attacks and a high anonymity level.

On the other hand, *low-latency* constraints seriously limit the anonymity that can be provided against powerful adversaries. Currently deployed solutions are vulnerable against attackers who have access to both ends of the communication. In particular, the variability of HTTP traffic makes it hard to conceal the correlation of input and output at the edges of the network using black box attacks. The approaches taken so far are to either use cascades with tight control on who uses the channel and how [114, 158] or to use free routes and

grow the network (together with a few other tricks, such as guard nodes) so that it will be difficult for an adversary to observe both ends of a circuit with significant probability [63]. Trust and trustworthiness may also play a role in determining resistance to end-to-end correlation attacks.

There has been a link between anonymous communications and *censorship resistance* research, as solutions for one problem have been applied to the other. More research is needed to determine whether anonymity is the best tactic to distribute content that may be censored, or whether it adds cost that may be limiting the distribution even further. This is also related to the usability, growth, and performance questions that complicate network design and deployment decisions [66].

Finally, anonymous communication networks can be subject to a wide range of attacks. The most popular attacker models is the global attacker (with access to all communication lines, passive or active); and attackers capable of controlling only a subset of the network. The former makes the most sense against a relatively small network where high-latency countermeasures are available. Against a large, low-latency network it is too strong (global) and potentially too weak (if nodes cannot perform any active attacks). For the latter it is more reasonable to assume an attacker that can observe only a subset of the network but can perform active attacks from (some of) the network it observes. The attacks against which anonymity networks are most vulnerable include traffic analysis, flooding, compulsion, and attacks on the cryptographic protocols (such as tagging attacks).

Know-how in attacking anonymous communication grows at the same, or even faster, rate as our ability to design secure systems. As more systems are deployed, further attacks are uncovered, attacks making use of the implementation environment and the actual usage of the anonymity systems. Anonymity design has proved to be a nontrivial problem, but so far we have only scraped the surface of the anonymity engineering, deployment and operations problems.

References

[1] Masayuki Abe. Universally verifiable mix-net with verification work independent of the number of mix-servers. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT '98*, pages 437–447, Helsinki, Finland, 1998. Springer-Verlag, LNCS 1403.

[2] Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the economics of anonymity. In Rebecca N. Wright, editor, *Financial Cryptography, 7th International Conference, FC 2003*, pages 84–102. Springer-Verlag, LNCS 2742, 2003.

[3] Ben Adida and Douglas Wikström. Obfuscated ciphertext mixing. Cryptology ePrint Archive, Report 2005/394, November 2005.

[4] Dakshi Agrawal, Dogan Kesdogan, and Stefan Penz. Probabilistic treatment of mixes to hamper traffic analysis. In *Proceedings, 2003 IEEE Symposium on Security and Privacy*, pages 16–27. IEEE Computer Society, May 2003.

[5] Ross Anderson. The eternity service. In *1st International Conference on the Theory and Applications of Cryptology (Pragocrypt '96)*, pages 242–252, Prague, Czech Republic, September/October 1996. Czech Technical University Publishing House.

[6] Dmitri Asonov and Johann-Christoph Freytag. Almost optimal private information retrieval. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 239–243, San Francisco, CA, USA, April 2002. Springer-Verlag, LNCS 2482.

[7] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In Catherine Meadows and Paul Syverson, editors, *CCS'05: Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 92–101. ACM Press, November 2005.

[8] Adam Back, Ian Goldberg, and Adam Shostack. Freedom systems 2.0 security issues and analysis. White paper, Zero Knowledge Systems, Inc., October 2001. The attributed date is that printed at the head of the paper. The cited work is, however, superceded by documents that came before Oct. 2001, e.g., [9].

[9] Adam Back, Ian Goldberg, and Adam Shostack. Freedom systems 2.1 security issues and analysis. White paper, Zero Knowledge Systems, Inc., May 2001.

[10] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Information Hiding: 4th International Workshop, IH 2001*, pages 245–257, Pittsburgh, PA, USA, April 2001. Springer-Verlag, LNCS 2137.

[11] Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Low-resource routing attacks against Tor. In Ting Yu, editor, *WPES'07: Proceedings of the 2007 ACM Workshop on Privacy in the Electronic Society*, pages 11–20. ACM Press, October 2007.

[12] Amos Beimel and Shlomi Dolev. Buses for anonymous message delivery. *Journal of Cryptology*, 16(1):25–39, 2003.

[13] Krista Bennett and Christian Grothoff. GAP – practical anonymous networking. In Roger Dingledine, editor, *Privacy Enhancing Technologies: Third International Workshop, PET 2003*, pages 141–160. Springer-Verlag, LNCS 2760, 2003.

[14] Ron Berman, Amos Fiat, and Amnon Ta-Shma. Provable unlinkability against traffic analysis. In Ari Juels, editor, *Financial Cryptography, 8th International Conference, FC 2004*, pages 266–289. Springer-Verlag, LNCS 3110, February 2004.

[15] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 115–129. Springer-Verlag, LNCS 2009, July 2000.

[16] Oliver Berthold and Heinrich Langos. Dummy traffic against long term intersection attacks. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 110–128, San Francisco, CA, USA, April 2002. Springer-Verlag, LNCS 2482.

[17] Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The disadvantages of free MIX routes and how to overcome them. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 30–45. Springer-Verlag, LNCS 2009, July 2000.

[18] John Bethencourt, Dawn Xiaodong Song, and Brent Waters. New constructions and practical applications for private stream searching (extended abstract). In *2006 IEEE Symposium on Security and Privacy (S&P 2006), Proceedings*, pages 132–139. IEEE Computer Society, May 2006.

[19] George Dean Bissias, Marc Liberatore, , and Brian Neil Levine. Privacy vulnerabilities in encrypted HTTP streams. In George Danezis and David Martin, editors, *Privacy Enhancing Technologies: 5th International Workshop, PET 2005*, Cavtat Croatia, 2005. Springer-Verlag, LNCS 3856.

[20] Avrim Blum, Dawn Xiaodong Song, and Shobha Venkataraman. Detection of interactive stepping stones: Algorithms and confidence bounds. In Erland Jonsson, Alfonso Valdes, and Magnus Almgren, editors, *Recent Advances in Intrusion Detection 7th International Symposium, RAID 2004*, pages 258–277, Sophia Antipolis, France, 2004. Springer-Verlag, LNCS 765.

[21] Nikita Borisov. An analysis of parallel mixing with attacker-controlled inputs. In George Danezis and David Martin, editors, *Privacy Enhancing Technologies: 5th International Workshop, PET 2005*, pages 12–25, Cavtat Croatia, 2005. Springer-Verlag, LNCS 3856.

[22] Nikita Borisov. *Anonymous Routing in Structured Peer-to-Peer Overlays*. PhD thesis, University of California, Berkeley, 2005.

[23] Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., December 2000.

[24] Justin Boyan. The Anonymizer: Protecting user privacy on the web. *CMC Magazine*, September 1997.

[25] Zach Brown. Cebolla: Pragmatic IP Anonymity. In *Proceedings of the 2002 Ottawa Linux Symposium*, June 2002.

[26] Jan Camenisch and Anna Lysyanskaya. A formal treatment of onion routing. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005: 25th Annual International Cryptology Conference*, pages 169–187. Springer-Verlag, LNCS 3621, August 2005.

[27] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2):84–88, February 1981.

[28] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology – CRYPTO '82*, pages 199–203, New York and London, 1983. Plenum Press.

[29] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.

[30] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[31] Heyning Cheng and Ron Avnur. Traffic analysis of ssl encrypted web browsing. `http://citeseer.ist.psu.edu/656522.html`.

[32] Giusseppe Ciaccio. Improving sender anonymity in a structured overlay with imprecise routing. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies: 6th International Workshop, PET 2006*, pages 190–207. Springer-Verlag, LNCS 4258, 2006.

[33] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66. Springer-Verlag, LNCS 2009, 2000.

[34] Richard Clayton. Failure in a hybrid content blocking system. In George Danezis and David Martin, editors, *Privacy Enhancing Technologies: 5th International Workshop, PET 2005*, pages 78–92, Cavtat Croatia, 2005. Springer-Verlag, LNCS 3856.

[35] Richard Clayton, Steven J. Murdoch, and Robert N. M. Watson. Ignoring the great firewall of china. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies: 6th International Workshop, PET 2006*, pages 20–35. Springer-Verlag, LNCS 4258, 2006.

[36] George Danezis. Traffic analysis of the HTTP protocol over TLS. `http://www.cl.cam.ac.uk/~gd216/TLSanon.pdf`.

[37] George Danezis. Forward secure mixes. In Jonsson Fisher-Hubner, editor, *Nordic workshop on Secure IT Systems (NordSec 2002)*, pages 195–207, Karlstad, Sweden, November 2002.

[38] George Danezis. Mix-networks with restricted routes. In Roger Dingledine, editor, *Privacy Enhancing Technologies: Third International Workshop, PET 2003*, pages 1–17. Springer-Verlag, LNCS 2760, 2003.

[39] George Danezis. Statistical disclosure attacks. In Gritzalis, Vimercati, Samarati, and Katsikas, editors, *Security and Privacy in the Age of Uncertainty, (SEC2003)*, pages 421–426, Athens, May 2003. IFIP TC11, Kluwer.

[40] George Danezis. The traffic analysis of continuous-time mixes. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies: 4th International Workshop, PET 2004*. Springer-Verlag, LNCS 3424, May 2005.

[41] George Danezis. Breaking four mix-related schemes based on universal re-encryption. In Sokratis K. Katsikas, Javier Lopez, Michael Backes, Stefanos Gritzalis, and Bart Preneel, editors, *Information Security 9th International Conference, ISC 2006*, pages 46–59, Samos Island, Greece, September 2006. Springer-Verlag, LNCS 4176.

[42] George Danezis and Richard Clayton. Route fingerprinting in anonymous communications. In *Sixth IEEE International Conference on Peer-to-Peer Computing, P2P 2006*, pages 69–72. IEEE Computer Society Press, 2006.

[43] George Danezis and Jolyon Clulow. Compulsion resistant anonymous communications. In Mauro Barni, Jordi Herrera-Joancomartí, Stefan Katzenbeisser, and Fernando Pérez-González, editors, *Information Hiding: 7th International Workshop, IH 2005*, pages 11–25. Springer-Verlag, LNCS 3727, June 2005.

[44] George Danezis and Claudia Diaz. Space-efficient private search with applications to rateless codes. In Sven Dietrich and Rachna Dahamija, editors, *Financial Cryptography and Data Security , 11th International Conference, FC 2007, and 1st International Workshop on Usable Security, USEC 2007*, pages 148–162. Springer-Verlag, LNCS 4886, 2007.

[45] George Danezis, Claudia Diaz, and Carmela Troncoso. Two-sided statistical disclosure attack. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies: 7th International Symposium, PET 2007*, pages 30–44. Springer-Verlag, LNCS 4776, 2007.

[46] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *Proceedings, IEEE Symposium on Security and Privacy*, pages 2–15, Berkeley, CA, May 2003. IEEE Computer Society.

[47] George Danezis and Ben Laurie. Minx: A simple and efficient anonymous packet format. In Sabrina De Capitani di Vimercati and Paul Syverson, editors, *WPES'04: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 59–65, Washington, DC, USA, October 2004. ACM Press.

[48] George Danezis and Len Sassaman. Heartbeat traffic to counter $(n-1)$ attacks. In Pierangela Samarati and Paul Syverson, editors, *WPES'03: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*, pages 89–93, Washington, DC, USA, October 2003. ACM Press.

[49] George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In Jessica Fridrich, editor, *Information Hiding: 6th International Workshop, IH 2004*, pages 293–308. Springer-Verlag, LNCS 3200, May 2004.

[50] George Danezis and Paul Syverson. Bridging and fingerprinting: Epistemic attacks on route selection. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies: Eighth International Symposium, PETS 2008*, pages 151–166. Springer-Verlag, LNCS 5134, July 2008.

[51] Yvo Desmedt and Kaoru Kurosawa. How to break a practical MIX and design a new one. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, pages 557–572, Bruges, Belgium, May 2000. Springer-Verlag, LNCS 1807.

[52] Claudia Diaz. *Anonymity and Privacy in Electronic Services*. PhD thesis, Katholieke Universiteit Leuven, 2005.

[53] Claudia Diaz, George Danezis, Christian Grothoff, Andreas Pfitzmann, and Paul F. Syverson. Panel discussion - mix cascades versus peer-to-peer: Is one concept superior? In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies: 4th International Workshop, PET 2004*, page 242. Springer-Verlag, LNCS 3424, 2005.

[54] Claudia Diaz and Bart Preneel. Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In Jessica Fridrich, editor, *Information Hiding: 6th International Workshop, IH 2004*, pages 309–325. Springer-Verlag, LNCS 3200, May 2004.

[55] Claudia Diaz, Len Sassaman, and Evelyne Dewitte. Comparison between two practical mix designs. In Pierangela Samarati, Peter Ryan, Dieter Gollmann, and Refik Molva, editors, *Computer Security – ESORICS 2004, 9th European Symposium on Research in Computer Security*, pages 141–159. Springer-Verlag, LNCS 3193, 2004.

[56] Claudia Diaz and Andrei Serjantov. Generalising mixes. In Roger Dingledine, editor, *Privacy Enhancing Technologies: Third International Workshop, PET 2003*, pages 18–31, Dresden, Germany, 2003. Springer-Verlag, LNCS 2760.

[57] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 54–68, San Francisco, CA, USA, April 2002. Springer-Verlag, LNCS 2482.

[58] Claudia Diaz, Carmela Troncoso, and George Danezis. Does additional information always reduce anonymity? In Ting Yu, editor, *WPES'07: Proceedings of the 2007 ACM Workshop on Privacy in the Electronic Society*, pages 72–75. ACM Press, October 2007.

[59] Claudia Diaz, Carmela Troncoso, and Andrei Serjantov. On the impact of social network profiling on anonymity. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies: Eighth International Symposium, PETS 2008*, pages 44–62. Springer-Verlag, LNCS 5134, July 2008.

[60] T. Dierks and C. Allen. RFC 2246: The TLS protocol version 1.0. `http://www.ietf.org/rfc/rfc2246.txt`, January 1999.

[61] Roger Dingledine, Michael J. Freedman, David Hopwood, and David Molnar. A reputation system to increase MIX-net reliability. In Ira S. Moskowitz, editor, *Information Hiding: 4th International Workshop, IH 2001*, pages 126–141, Pittsburgh, PA, USA, April 2001. Springer-Verlag, LNCS 2137.

[62] Roger Dingledine, Michael J. Freedman, and David Molnar. The free haven project: Distributed anonymous storage service. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 67–95. Springer-Verlag, LNCS 2009, 2000.

[63] Roger Dingledine and Nick Mathewson. Anonymity loves company: Usability and the network effect. In Ross Anderson, editor, *Fifth Workshop on the Economics of Information Security (WEIS 2006)*, June 2006.

[64] Roger Dingledine and Nick Mathewson. Design of a blocking-resistant anonymity system (draft). `https://www.torproject.org/svn/trunk/doc/design-paper/blocking.html`, May 2007.

[65] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–319. USENIX Association, August 2004.

[66] Roger Dingledine, Nick Mathewson, and Paul Syverson. Deploying low-latency anonymity: Design challenges and social factors. *IEEE Security & Privacy*, 5(5):83–87, September/October 2007.

[67] Roger Dingledine, Andrei Serjantov, and Paul Syverson. Blending different latency traffic with alpha-mixing. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies: 6th International Workshop, PET 2006*, pages 245–257. Springer-Verlag, LNCS 4258, 2006.

[68] Roger Dingledine, Vitaly Shmatikov, and Paul Syverson. Synchronous batching: From cascades to free routes. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies: 4th International Workshop, PET 2004*, pages 186–206. Springer-Verlag, LNCS 3424, May 2005.

[69] Roger Dingledine and Paul Syverson. Reliable mix cascade networks through reputation. In Matt Blaze, editor, *Financial Cryptography, 6th International Conference, FC 2002*, pages 253–268. Springer-Verlag, LNCS 2357, 2003.

[70] M. Dornseif. Government mandated blocking of foreign web content. In J. von Knop, W. Haverkamp, and E. Jessen, editors, *Security, E-Learning, E-Services: Proceedings of the 17. DFN-Arbeitstagung uber Kommunikationsnetze, Dusseldorf*, 2003.

[71] John Douceur. The sybil attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Peer-To-Peer Systems: First International Workshop, IPTPS 2002*, pages 251–260, Cambridge, MA, USA, 2002. Springer-Verlag, LNCS 2429.

[72] Matthew Edman, Fikret Sivrikaya, and Bülent Yener. A combinatorial approach to measuring anonymity. In Gheorghe Muresan, Tayfur Altiok, Benjamin Melamed, and Daniel Zeng, editors, *IEEE Intelligence and Security Informatics (ISI 2007)*, pages 356–363, New Brunswick, NJ, May 2007. IEEE.

[73] Matthew Edman and Paul Syverson. AS-awareness in Tor path selection. In Somesh Jha, Angelos D. Keromytis, and Hao Chen, editors, *CCS'09: Proceedings of the 16th ACM Conference on Computer and Communications Security*. ACM Press, 2009.

[74] Nathan S. Evans, Roger Dingledine, and Christian Grothoff. A practical congestion attack on Tor using long paths. In *Proceedings of the 18th USENIX Security Symposium*, pages 33–50, Montreal, Canada, August 2009. USENIX Association.

[75] Peter Fairbrother. An improved construction for universal re-encryption. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies: 4th International Workshop, PET 2004*, pages 79–87. Springer-Verlag, LNCS 3424, May 2005.

[76] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing web censorship and surveillance. In Dan Boneh, editor, *USENIX Security Symposium*, pages 247–262, San Francisco, CA, 5-9 August 2002.

[77] Nick Feamster, Magdalena Balazinska, Winston Wang, Hari Balakrishnan, and David Karger. Thwarting web censorship with untrusted messenger discovery. In Roger Dingledine, editor, *Privacy Enhancing Technologies: Third International Workshop, PET 2003*, pages 125–140. Springer-Verlag, LNCS 2760, 2003.

[78] Nick Feamster and Roger Dingledine. Location diversity in anonymity networks. In Sabrina De Capitani di Vimercati and Paul Syverson, editors, *WPES'04: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 66–76, Washington, DC, USA, October 2004. ACM Press.

[79] Joan Feigenbaum, Aaron Johnson, and Paul Syverson. Probabilistic analysis of onion routing in a black-box model [extended abstract]. In Ting Yu, editor, *WPES'07: Proceedings of the 2007 ACM Workshop on Privacy in the Electronic Society*, pages 1–10. ACM Press, October 2007.

[80] Laurent Fousse and Jean-Ren Reinhard. Nymbaron: A type iii nymserver. On-line, 2006. `http://www.komite.net/laurent/soft/nymbaron/`.

[81] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In Vijay Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002*, pages 193–206. ACM Press, 2002.

[82] Michael J. Freedman, Emil Sit, Josh Cates, and Robert Morris. Introducing tarzan, a peer-to-peer anonymizing network layer. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Peer-To-Peer Systems: First International Workshop, IPTPS 2002*, pages 121–129, Cambridge, MA, USA, 2002. Springer-Verlag, LNCS 2429.

[83] Jun Furukawa and Kazue Sako. An efficient scheme for proving a shuffle. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, pages 368–387, Santa Barbara, CA, USA, August 2001. Springer-Verlag, LNCS 2139.

[84] Benedikt Gierlichs, Carmela Troncoso, Claudia Diaz, Bart Preneel, and Ingrid Verbauwhede. Revisiting a combinatorial approach toward measuring anonymity. In Marianne Winslett, editor, *WPES'08: Proceedings*

of the 2008 ACM Workshop on Privacy in the Electronic Society, pages 111–116, Alexandria,VA,USA, October 2008. ACM Press.

[85] Sharad Goel, Mark Robson, Milo Polte, and Emin Gun Sirer. Herbivore: A scalable and efficient protocol for anonymous communication. Technical Report 2003-1890, Cornell University, Ithaca, NY, February 2003.

[86] Ian Goldberg. *A Pseudonymous Communications Infrastructure for the Internet*. PhD thesis, University of California at Berkeley, 2000.

[87] Ian Goldberg and Adam Shostack. Freedom 1.0 security issues and analysis. White paper, Zero Knowledge Systems, Inc., November 1999.

[88] Ian Goldberg and Adam Shostack. Freedom network 1.0 architecture and protocols. White paper, Zero Knowledge Systems, Inc., October 2001. The attributed date is that printed at the head of the paper. The cited work is, however, superceded by documents that came before Oct. 2001. The appendix indicates a change history with changes last made November 29, 1999. Also, in [87] the same authors refer to a paper with a similar title as an "April 1999 whitepaper".

[89] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In Ross Anderson, editor, *Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, 1996.

[90] Philippe Golle. Reputable mix networks. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies: 4th International Workshop, PET 2004*, pages 51–62. Springer-Verlag, LNCS 3424, May 2005.

[91] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In Tatsuaki Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, pages 163–178, San Francisco, USA, February 2004. Springer-Verlag, LNCS 2964.

[92] Philippe Golle and Ari Juels. Dining cryptographers revisited. In *Advances in Cryptology – EUROCRYPT 2004*, pages 456–473, Interlaken, Switzerland, May 2004. Springer-Verlag, LNCS 3027.

[93] Philippe Golle and Ari Juels. Parallel mixing. In Birgit Pfitzmann and Peng Liu, editors, *CCS 2004: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 220–226. ACM Press, October 2004.

[94] Philippe Golle, XiaoFeng Wang, Markus Jakobsson, and Alex Tsow. Deterring voluntary trace disclosure in re-encryption mix networks. In *2006 IEEE Symposium on Security and Privacy (S& P 2006), Proceedings*, pages 121–131, Oakland, CA, May 2006. IEEE Computer Society.

[95] Philippe Golle, Sheng Zhong, Dan Boneh, Markus Jakobsson, and Ari Juels. Optimistic mixing for exit-polls. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, pages 451–465, Queenstown, New Zealand, 1-5 December 2002. Springer-Verlag, LNCS 2501.

[96] Marcin Gomułkiewicz, Marek Klonowski, and Mirosław Kutyłowski. Onions based on universal re-encryption – anonymous communication immune against repetitive attack. In Chae Hoon Lim and Moti Yung, editors, *Information Security Applications, 5th International Workshop, WISA 2004*, pages 400–410, Jeju Island, Korea, August 2004. Springer-Verlag, LNCS 3325.

[97] Marcin Gomułkiewicz, Marek Klonowski, and Mirosław Kutyłowski. Rapid mixing and security of chaum's visual electronic voting. In Einar Snekkenes and Dieter Gollmann, editors, *Computer Security – ESORICS 2003, 8th European Symposium on Research in Computer Security*, pages 132–145, Gjøvik, Norway, October 2003. Springer-Verlag, LNCS 2808.

[98] Yong Guan, Xinwen Fu, Dong Xuan, P. U. Shenoy, Riccardo Bettati, and Wei Zhao. Netcamo: camouflaging network traffic for qos-guaranteed mission critical applications. *IEEE Transactions on Systems, Man, and Cybernetics*, Part A 31(4):253–265, 2001.

[99] Ceki Gülcü and Gene Tsudik. Mixing E-mail with Babel. In *Proceedings of the Symposium on Network and Distributed Security Symposium - NDSS '96*, pages 2–16. IEEE, February 1996.

[100] Joseph Y. Halpern and Kevin R. O'Neill. Anonymity and information hiding in multiagent systems. *Jounal of Computer Security*, 13(3):483–514, 2005.

[101] Steven Hazel and Brandon Wiley. Achord: A variant of the chord lookup service for use in censorship resistant peer-to-peer publishing systems. In *Peer-To-Peer Systems: First International Workshop, IPTPS 2002*, Cambridge, MA, USA, 2002. A postproceedings of this workshop was published by Springer-Verlag (LNCS 2429). This paper is not in that volume. It is only in the electronic proceedings available at `http://www.iptps.org/papers.html#2002`.

[102] Sabine Helmers. A brief history of anon.penet.fi - the legendary anonymous remailer. *CMC Magazine*, September 1997.

[103] Johan Helsingius. Johan Helsingius closes his internet remailer. `http://www.penet.fi/press-english.html`, August 1996.

[104] Johan Helsingius. Temporary injunction in the anonymous remailer case. `http://www.penet.fi/injuncl.html`, September 1996.

[105] Alejandro Hevia and Daniele Micciancio. An indistinguishability-based characterization of anonymous channels. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies: Eighth International Symposium, PETS 2008*, pages 24–43. Springer-Verlag, LNCS 5134, July 2008.

[106] Andrew Hintz. Fingerprinting websites using traffic analysis. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 171–178, San Francisco, CA, USA, April 2002. Springer-Verlag, LNCS 2482.

[107] Nicholas Hopper, Eugene Y. Vasserman, and Eric Chan-Tin. How much anonymity does network latency leak? In Sabrina De Capitani di Vimercati, Paul Syverson, and David Evans, editors, *CCS'07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 82–91. ACM Press, 2007.

[108] Dominic Hughes and Vitaly Shmatikov. Information hiding, anonymity and privacy: A modular approach. *Journal of Computer Security*, 12(1):3–36, 2004.

[109] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 239–248, Washington, DC, USA, 2006. IEEE Computer Society.

[110] Markus Jakobsson. A practical mix. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT '98*, pages 448–461, Helsinki, Finland, 1998. Springer-Verlag, LNCS 1403.

[111] Markus Jakobsson. Flash mixing. In *PODC '99: Proceedings of the Eighteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 83–89, Atlanta, Georgia, USA, 1999. ACM Press.

[112] Markus Jakobsson and Ari Juels. An optimally robust hybrid mix network. In *PODC '01: Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 284–292, New York, NY, USA, 2001. ACM.

[113] Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In Dan Boneh, editor, *Proceedings of the 11th USENIX Security Symposium*, pages 339–353, San Francisco, CA, USA, 5-9 August 2002. USENIX Association.

[114] Anja Jerichow, Jan Müller, Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. Real-time MIXes: A bandwidth-efficient anonymity protocol. *IEEE Journal on Selected Areas in Communications*, 16(4):495–509, May 1998.

[115] Shu Jiang, Nitin H. Vaidya, and Wei Zhao. Routing in packet radio networks to prevent traffic analsyis. In *IEEE Information Assurance and Security Workshop*, June 2000.

[116] Aaron Johnson and Paul Syverson. More anonymous onion routing through trust. In *22nd IEEE Computer Security Foundations Symposium, CSF 2009*, pages 3–12, Port Jefferson, New York, July 2009. IEEE Computer Society.

[117] Aniket Kate, Greg Zaverucha, and Ian Goldberg. Pairing-based onion routing. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies: 7th International Symposium, PET 2007*, pages 95–112. Springer-Verlag, LNCS 4776, 2007.

[118] Sachin Katti, Dina Katabi, and Katay Puchala. Slicing the onion: Anonymous routing without pki. In *Fourth Workshop on Hot Topics in Networks (HotNets-IV)*. ACM, 2005. `http://conferences.sigcomm.org/hotnets/2005/papers/katti.pdf`.

[119] Dogan Kesdogan, Dakshi Agrawal, and Stefan Penz. Limits of anonymity in open environments. In Fabien A. P. Petitcolas, editor, *Information Hiding: 5th International Workshop, IH 2002*, pages 53–69, Noordwijkerhout, The Netherlands, October 2002. Springer-Verlag, LNCS 2578.

[120] Dogan Kesdogan, Mark Borning, and Michael Schmeink. Unobservable surfing on the world wide web: Is private information retrieval an alternative to the MIX based approach? In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 214–218, San Francisco, CA, USA, April 2002. Springer-Verlag, LNCS 2482.

[121] Dogan Kesdogan, Jan Egner, and Roland Büschkes. Stop-and-Go MIXes: Providing probabilistic anonymity in an open system. In David Aucsmith, editor, *Information Hiding: Second International Workshop, IH 1998*, pages 83–98, Portland, Oregon, USA, April 1998. Springer-Verlag, LNCS 1525.

[122] Joe Kilian and Kazue Sako. Receipt-free MIX-type voting scheme — a practical solution to the implementation of a voting booth. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology – EUROCRYPT '95*, pages 393–403, Saint-Malo, France, May 1995. Springer-Verlag, LNCS 921.

[123] Lea Kissner, Alina Oprea, Michael K. Reiter, Dawn Xiaodong Song, and Ke Yang. Private keyword-based push and pull with applications to anonymous communication. In Markus Jakobsson, MotiYung, and Jianying Zhou, editors, *Applied Cryptography and Network Security Second International Conference, ACNS 2004*, pages 16–30. Springer-Verlag, LNCS 3089, 2004.

[124] Marek Klonowski, Mirosław Kutyłlowski, Anna Lauks, and Filip Zagórski. Universal re-encryption of signatures and controlling anonymous information flow. In *WARTACRYPT '04 Conference on Cryptology*, Bedlewo/Poznan, July 2004.

[125] Marek Klonowski, Mirosław Kutyłlowski, and Filip Zagórski. Anonymous communication with on-line and off-line onion encoding. In Peter Vojtáš, Mária Bieliková, Bernadette Charron-Bost, and Ondrej Sýkora, editors, *SOFSEM 2005: Theory and Practice of Computer Science, 31st Conference on Current Trends in Theory and Practice of Computer Science*, Lecture Notes in Computer Science, pages 229–238, Liptovský Ján, Slovakia, January 2005. Springer-Verlag, LNCS 3381.

[126] Marek Klonowski and Mirosław Kutyłowski. Provable anonymity for networks of mixes. In Mauro Barni, Jordi Herrera-Joancomartí, Stefan Katzenbeisser, and Fernando Pérez-González, editors, *Information Hiding: 7th International Workshop, IH 2005*, pages 26–38. Springer-Verlag, LNCS 3727, June 2005.

[127] Stefan Köpsell and Ulf Hilling. How to achieve blocking resistance for existing systems enabling anonymous web surfing. In Sabrina De Capitani di Vimercati and Paul Syverson, editors, *WPES'04: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 47–58, Washington, DC, USA, October 2004. ACM Press.

[128] Dennis Kügler. An analysis of GNUnet and the implications for anonymous, censorship-resistant networks. In Roger Dingledine, editor, *Privacy Enhancing Technologies: Third International Workshop, PET 2003*, pages 161–176. Springer-Verlag, LNCS 2760, 2003.

[129] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew K. Wright. Timing attacks in low-latency mix-based systems. In Ari Juels, editor, *Financial Cryptography, 8th International Conference, FC 2004*, pages 251–265. Springer-Verlag, LNCS 3110, February 2004.

[130] Brian Neil Levine and Clay Shields. Hordes: a multicast based protocol for anonymity. *Journal of Computer Security*, 10(3):213–240, 2002.

[131] John Leyden. Anonymizer looks for gaps in great firewall of China. The Register, April 3 2006.

[132] Tianbo Lu, Binxing Fang, Yuzhong Sun, and Xueqi Cheng. Performance analysis of WonGoo system. In *Fifth International Conference on Computer and Information Technology (CIT 2005)*, pages 716–723, Shanghai, China, September 2005. IEEE Computer Society.

[133] Tianbo Lu, Binxing Fang, Yuzhong Sun, and Li Guo. Some remarks on universal re-encryption and a novel practical anonymous tunnel. In Xicheng Lu and Wei Zhao, editors, *Networking and Mobile Computing,*

Third International Conference, ICCNMC 2005, pages 853–862, Zhangjiajie, China, 2005. Springer-Verlag, LNCS 3619.

[134] David Martin and Andrew Schulman. Deanonymizing users of the SafeWeb anonymizing service. Technical Report 2002-003, Boston University Computer Science Department, February 2002.

[135] Nick Mathewson. Underhill: A proposed type 3 nymserver protocol specification. On-line, 2005. `http://svn.conuropsis.org/nym3/trunk/doc/nym-spec.txt`.

[136] Nick Mathewson and Roger Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies: 4th International Workshop, PET 2004*. Springer-Verlag, LNCS 3424, 2005.

[137] David Mazières and M. Frans Kaashoek. The Design, Implementation and Operation of an Email Pseudonym Server. In *CCS'98 – 5th ACM Conference on Computer and Communications Security*, pages 27–36, San Francisco, CA, USA, November 1998. ACM Press.

[138] Markus Michels and Patrick Horster. Some remarks on a receipt-free and universally verifiable mix-type voting scheme. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology – ASIACRYPT '96*, pages 125–132, Kyongju, Korea, November 1996. Springer-Verlag, LNCS 1163.

[139] Masashi Mitomo and Kaoru Kurosawa. Attack for flash MIX. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, pages 192–204, Kyoto, Japan, December 2000. Springer-Verlag, LNCS 1976.

[140] Prateek Mittal and Nikita Borisov. Information leaks in structured peer-to-peer anonymous communication systems. In Paul Syverson, Somesh Jha, and Xiaolan Zhang, editors, *CCS'08: Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 267–278. ACM Press, 2008.

[141] Bodo Möller. Provably secure public-key encryption for length-preserving chaumian mixes. In Marc Joye, editor, *Topics in Cryptology – CT-RSA 2003*, pages 244–262, San Francisco, CA, USA, 13-17 April 2003. Springer-Verlag, LNCS 2612.

[142] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster protocol - version 3. IETF Internet Draft, 2003.

[143] Steven J. Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *2005 IEEE Symposium on Security and Privacy,(IEEE S&P 2005) Proceedings*, pages 183–195. IEEE CS, May 2005.

[144] Steven J. Murdoch and Piotr Zieliński. Sampled traffic analysis by internet-exchange-level adversaries. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies: 7th International Symposium, PET 2007*, pages 167–183. Springer-Verlag, LNCS 4776, 2007.

[145] Arjun Nambiar and Matthew Wright. Salsa: A structured approach to large-scale anonymity. In Rebecca N. Wright, Sabrina De Capitani di Vimercati, and Vitaly Shmatikov, editors, *CCS'06: Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 17–26. ACM Press, 2006.

[146] C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In Pierangela Samarati, editor, *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS-8)*, pages 116–125, Philadelphia, PA, USA, November 2001. ACM Press.

[147] Richard E. Newman, Ira S. Moskowitz, Paul Syverson, and Andrei Serjantov. Metrics for traffic analysis prevention. In Roger Dingledine, editor, *Privacy Enhancing Technologies: Third International Workshop, PET 2003*, pages 48–65, Dresden, Germany, March 2003. Springer-Verlag, LNCS 2760.

[148] Luke O'Connor. On blending attacks for mixes with memory. In Mauro Barni, Jordi Herrera-Joancomartí, Stefan Katzenbeisser, and Fernando Pérez-González, editors, *Information Hiding: 7th International Workshop, IH 2005*, pages 39–52. Springer-Verlag, LNCS 3727, June 2005.

[149] Wakaha Ogata, Kaoru Kurosawa, Kazue Sako, and Kazunori Takatani. Fault tolerant anonymous channel. In Yongfei Han, Tatsuaki Okamoto, and Sihan Qing, editors, *Information and Communication Security, First International Conference, ICICS '97*, pages 440–444, Beijing, China, November 1997. Springer-Verlag, LNCS 1334.

[150] Miyako Ohkubo and Masayuki Abe. A length-invariant hybrid mix. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, pages 178–191, Kyoto, Japan, December 2000. Springer-Verlag, LNCS 1976.

[151] Rafail Ostrovsky and William E. Skeith III. Private searching on streaming data. In *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference*, pages 223–240. Springer-Verlag, LNCS 3621, 2005.

[152] Lasse Øverlier and Paul Syverson. Locating hidden servers. In *2006 IEEE Symposium on Security and Privacy (S& P 2006), Proceedings*, pages 100–114. IEEE CS, May 2006.

[153] Lasse Øverlier and Paul Syverson. Improving efficiency and simplicty of Tor circuit establishment and hidden services. In Nikita Borisov and

Philippe Golle, editors, *Privacy Enhancing Technologies: 7th International Symposium, PET 2007*, pages 134–152. Springer-Verlag, LNCS 4776, 2007.

[154] Peter Palfrader. Echolot: a pinger for anonymous remailers. `http://www.palfrader.org/echolot/`.

[155] Sameer Parekh. Prospects for remailers: where is anonymity heading on the internet? *First Monday*, 1(2), August 5 1996. On-line journal `http://www.firstmonday.dk/issues/issue2/remailers/`.

[156] C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT '93*, pages 248–259. Springer-Verlag, LNCS 765, 1994.

[157] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity – a proposal for terminology. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 1–9. Springer-Verlag, LNCS 2009, July 2000.

[158] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. ISDN-MIXes: Untraceable communication with very small bandwidth overhead. In Wolfgang Effelsberg, Hans Werner Meuer, and Günter Müller, editors, *Kommunikation in Verteilten Systemen, Grundlagen, Anwendungen, Betrieb, GI/ITG-Fachtagung*, volume 267 of *Informatik-Fachberichte*, pages 451–463. Springer-Verlag, February 1991.

[159] Andreas Pfitzmann and Michael Waidner. Networks without user observability – design options. In Franz Pichler, editor, *Advances in Cryptology – EUROCRYPT '85*, pages 245–253. Springer-Verlag, LNCS 219, 1986.

[160] Birgit Pfitzmann. Breaking an efficient anonymous channel. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT '94*, pages 332–340, Perugia, Italy, May 1994. Springer-Verlag, LNCS 950.

[161] Birgit Pfitzmann and Andreas Pfitzmann. How to break the direct RSA-implementation of MIXes. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology – EUROCRYPT '89*, pages 373–381, Houthalen, Belgium, April 1990. Springer-Verlag, LNCS 434.

[162] Charles Rackoff and Daniel R. Simon. Cryptographic defense against traffic analysis. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 672–681, New York, NY, USA, 1993. ACM.

[163] Josyula R. Rao and Pankaj Rohatgi. Can pseudonymity really guarantee privacy? In *Proceedings of the 9th USENIX Security Symposium*, pages 85–96. USENIX Association, August 2000.

[164] Jean-François Raymond. Traffic analysis: Protocols, attacks, design issues, and open problems. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.

[165] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998.

[166] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998.

[167] Michael Reiter and XiaoFeng Wang. Fragile mixing. In Birgit Pfitzmann and Peng Liu, editors, *CCS 2004: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 227–235. ACM Press, October 2004.

[168] Marc Rennhard and Bernhard Plattner. Introducing MorphMix: Peer-to-peer based anonymous internet usage with collusion detection. In Sabrina De Capitani di Vimercati and Pierangela Samarati, editors, *Proceedings of the ACM Workshop on Privacy in the Electronic Society, WPES 2002*, pages 91–102. ACM Press, 2002.

[169] Len Sassaman, Bram Cohen, and Nick Mathewson. The Pynchon Gate: A secure method of pseudonymous mail retrieval. In Sabrina De Capitani di Vimercati and Roger Dingledine, editors, *WPES'05: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, pages 1–9. ACM Press, October 2005.

[170] Andrei Serjantov. Anonymizing censorship resistant systems. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Peer-To-Peer Systems: First International Workshop, IPTPS 2002*, pages 111–120, Cambridge, MA, USA, 2002. Springer-Verlag, LNCS 2429.

[171] Andrei Serjantov. *On the Anonymity of Anonymity Systems*. PhD thesis, University of Cambridge, 2004.

[172] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies: Second International Workshop, PET 2002*, pages 41–53, San Francisco, CA, USA, April 2002. Springer-Verlag, LNCS 2482.

[173] Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In Fabien A.P. Petitcolas, editor, *Information Hiding: 5th International Workshop, IH 2002*, pages 36–52. Springer-Verlag, LNCS 2578, 2002.

[174] Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. In Einar Snekkenes and Dieter Gollmann, editors, *Computer Security – ESORICS 2003, 8th European Symposium on Research in Computer Security*, pages 141–159, Gjøvik, Norway, October 2003. Springer-Verlag, LNCS 2808.

[175] Claude Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

[176] Rob Sherwood, Bobby Bhattacharjee, and Aravind Srinivasan. P5: A protocol for scalable anonymous communication. In *Proceedings, 2002 IEEE Symposium on Security and Privacy*, pages 58–72, Berkeley, California, USA, May 2002. IEEE Computer Society.

[177] Erik Shimshock, Matt Staats, and Nick Hopper. Breaking and provably fixing minx. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies: Eighth International Symposium, PETS 2008*, pages 99–114. Springer-Verlag, LNCS 5134, July 2008.

[178] Vitaly Shmatikov. Probabilistic analysis of anonymity. In *15th IEEE Computer Security Foundations Workshop, CSFW-15*, pages 119–128, Cape Breton, Nova Scotia, Canada, June 2002. IEEE Computer Society.

[179] Michael Singer. CIA Funded SafeWeb Shuts Down. Internet News. `http://siliconvalley.internet.com/news/article.php/3531_926921`, November 20 2001.

[180] Qixiang Sun, Daniel R. Simon, Yi-Min Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings, IEEE Symposium on Security and Privacy*, pages 19–30. IEEE Computer Society, 2002.

[181] Paul Syverson. Why I'm not an entropist. In *Seventeenth International Workshop on Security Protocols*. Springer-Verlag, LNCS, 2009. Forthcoming.

[182] Paul Syverson, Michael Reed, and David Goldschlag. Onion Routing access configurations. In *Proceedings DARPA Information Survivability Conference & Exposition, DISCEX'00*, volume 1, pages 34–40. IEEE CS Press, 1999.

[183] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an analysis of onion routing security. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.

[184] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous connections and onion routing. In *Proceedings, 1997 IEEE Symposium on Security and Privacy*, pages 44–54. IEEE CS Press, May 1997.

[185] Paul F. Syverson and Stuart G. Stubblebine. Group principals and the formalization of anonymity. In Jeannette M. Wing, Jim Woodcock, and Jim Davies, editors, *FM'99 – Formal Methods, Vol. I*, pages 814–833. Springer-Verlag, LNCS 1708, September 1999.

[186] Parisa Tabriz and Nikita Borisov. Breaking the collusion detection mechanism of morphmix. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies: 6th International Workshop, PET 2006*, pages 368–383. Springer-Verlag, LNCS 4258, 2006.

[187] Brenda Timmerman. A security model for dynamic adaptive traffic masking. In *NSPW '97: Proceedings of the 1997 workshop on New security paradigms*, pages 107–116, New York, NY, USA, 1997. ACM.

[188] Brenda Timmerman. Secure dynamic adaptive traffic masking. In *NSPW '99: Proceedings of the 1999 workshop on New security paradigms*, pages 13–24, New York, NY, USA, 2000. ACM.

[189] Carmela Troncoso, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. Perfect matching disclosure attacks. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies: Eighth International Symposium, PETS 2008*, pages 2–23. Springer-Verlag, LNCS 5134, July 2008.

[190] B. R. Venkatraman and Richard E. Newman-Wolfe. Performance analysis of a method for high level prevention of traffic analysis using measurements from a campus network. In *Tenth Annual Computer Security Applications Conference*, pages 288–297, Orlando, FL, December 1994. IEEE CS Press.

[191] Michael Waidner and Birgit Pfitzmann. The dining cryptographers in the disco — underconditional sender and recipient untraceability with computationally secure serviceability. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology – EUROCRYPT '89*, page 690. Springer-Verlag, LNCS 434, 1990.

[192] G. Walton. Chinas golden shield: corporations and the development of surveillance technology in the Peoples Republic of China. *Montreal: International Centre for Human Rights and Democratic Development, URL (consulted 29 October 2001):* `http://www.ichrdd.ca/frame.iphtml`, 2001.

[193] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. Tracking anonymous peer-to-peer voip calls on the internet. In Catherine Meadows and Paul Syverson, editors, *CCS'05: Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 81–91. ACM Press, November 2005.

[194] Xinyuan Wang and Douglas S. Reeves. Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. In Vijay Atluri and Peng Liu, editors, *CCS 2003: Proceedings of the 10th ACM Conference on Computer and Communications Security*, pages 20–29, Washington, DC, USA, 2003.

[195] Douglas Wikström. How to break, fix, and optimize "optimistic mix for exit-polls". Technical Report T2002-24, Swedish Institute of Computer Science, SICS, Box 1263, SE-164 29 Kista, SWEDEN, 2002.

[196] Douglas Wikström. Elements in $Z_p^* \backslash G_q$ are dangerous. Technical Report T2003-05, Swedish Institute of Computer Science, SICS, Box 1263, SE-164 29 Kista, SWEDEN, 2003.

[197] Douglas Wikström. Four practical attacks for "optimistic mixing for exit-polls". Technical Report T2003-04, Swedish Institute of Computer Science, SICS, Box 1263, SE-164 29 Kista, SWEDEN, 2003.

[198] Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. An analysis of the degradation of anonymous protocols. In *Network and Distributed Security Symposium (NDSS '02)*, San Diego, California, 6-8 February 2002. Internet Society.

[199] Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. Defending anonymous communication against passive logging attacks. In *Proceedings, 2003 IEEE Symposium on Security and Privacy*, pages 28–43. IEEE Computer Society, May 2003.

[200] Ye Zhu and Riccardo Bettati. Un-mixing mix traffic. In George Danezis and David Martin, editors, *Privacy Enhancing Technologies: 5th International Workshop, PET 2005*, Cavtat Croatia, 2005. Springer-Verlag, LNCS 3856.

[201] Ye Zhu, Xinwen Fu, Bryan Graham, Riccardo Bettati, and Wei Zhao. On flow correlation attacks and countermeasures in mix networks. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies: 4th International Workshop, PET 2004*. Springer-Verlag, LNCS 3424, May 2005.