# Privacy Enhancing Technologies for the Internet III: Ten Years Later

Ian Goldberg
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON
iang@cs.uwaterloo.ca

## 1 Introduction

In 1997 with Wagner and Brewer, and again in 2002, we looked at the then-current state of privacy-enhancing technologies (PETs) for the Internet. [27, 26] Now, in 2007, we take a third look. Technologies to help users maintain their privacy online are as important today as ever before—if not more so. Identity theft is the fastest-growing crime in the US today [47] and it is all too easy for would-be identity thieves to harvest personal information from the online trails Internet users leave every day. Losses of large databases of personal information are an almost daily occurrence [2]; for example, retailers' servers are penetrated [44], databases are traded between government and private companies [36] and laptops containing social security numbers are stolen [35].

In 1997, we discussed the *dossier effect*: all available information about a person gets cross-referenced, and the resulting dossier ends up being used for many purposes, lawful and not. This practice has expanded over the years; the companies that compile and sell these dossiers are known as *data brokers*. Choicepoint is a prime example—in 2005, this data broker sold dossiers on over 150,000 Americans to a group of criminals. [10] The PETs we discuss here give people a way to control how much of their personal information is revealed when they use the Internet. By controlling the spread of this information, they can limit the size of the data brokers' dossiers about them.

In this article, we examine different classes of privacy-enhancing technologies. For each class, we look at the state of the technology in 2002 and see what has happened in the intervening five years. In section 2, we look at a range of systems to protect the identities of senders and recipients of electronic mail. In section 3, we examine systems which attempt the solve the more complex problem of protecting your identity when accessing interactive Internet services. Section 4 surveys a number of technologies which protect the contents of Internet conversations, as opposed to the identities of the participants. In section 5, we look to the future, and examine three particular technologies in which we hope to see progress in the next five years. Section 6 outlines the principles researchers should keep in mind when designing future security and privacy technologies in order to maximize their usefulness, and section 7 concludes.

# 2 Email anonymity and pseudonymity systems

The first class of PETs we will examine are systems to provide *anonymity* and *pseudonymity* for electronic mail. Email anonymity systems allow a user to *send* email without revealing his or her own personal information, such as identity, email address or IP address. Email pseudonymity systems also allow the user to set up a persistent pseudonym, or *nym*, which can be used to *receive* email as well. With these pseudonymous systems, users can participate in ongoing email conversations while maintaining their privacy.

## 2.1 Type-0 remailers

The oldest and simplest email anonymity systems were the *type-0 remailers*. The term *remailer* stems from the basic operation of these systems: a user sends email to the remailer, which strips off the user's identifying information and re-mails the message to its intended recipient. The remailer also assigns a random pseudonym to the sender. By keeping a master list matching the pseudonyms to senders' real email addresses, replies to remailed messages can be delivered to the original sender.

While these type-0 remailers provided some amount of protection against casual observers, the master list provided a tempting target for attackers; anyone who could get his hands on the list could reveal the real email addresses of all the users of the remailer. The most well-known of these remailers, `anon.penet.fi`, was shut down after its operator lost a legal fight that required him to turn over parts of the list. [30]

## 2.2 Type-I remailers

In order to better protect the privacy of email users, the *type-I*, or *cypherpunk remailers* were developed. They work on the same principle—a message arrives at a type-I remailer, which removes the sender's identifying information and then sends the message out. But these remailers add a number of key improvements. The first is *chaining*: a user sends his message to a remailer with instructions to send it, not to the intended recipient, but rather to a second remailer (run by an operator independent from the first). *That* remailer is instructed to send it to a third remailer, and so on. Only the last remailer in the chain receives the email address of the intended final recipient. Therefore, compromising any remailer or its operator does not allow linking the sender to the recipient. The first remailer knows only that the sender is a user of the remailer network, but not with whom he is communicating. The last remailer in the chain knows that somebody sent an anonymous message to a particular recipient, but cannot identify who. Remailers in the middle of the chain know only that they are forwarding anonymous email, but do not know the sender or recipient. The goal is that *all* of the remailers in the chain need to be compromised for the privacy of the sender to be breached.

The second improvement made by the type-I remailers is *encryption*. Without encryption, the first remailer in the chain could simply read the instructions to the later remailers, including the address of the final

recipient. Instead, the first remailer receives an encrypted message. When it decrypts it, it finds only the address of the second remailer and another encrypted message. This inner message, however, is encrypted to the *second* remailer, so the first remailer cannot read it. The first remailer sends that message to the second remailer, which decrypts it to find the address of the third remailer and another encrypted message (that only the third remailer can read), and so on. Finally, when the last remailer decrypts its message, it finds the address of the final recipient, as well as the (unencrypted) message to send.

The third improvement made by the type-I remailers is *mixing*: incoming messages to any remailer are batched together and randomly reordered before being sent out. This was to attempt to prevent a passive observer of a given remailer from determining which outgoing message corresponds to which incoming message. An attacker could perform a *timing correlation attack* by comparing the order in which messages were received by the remailer to the order in which they were subsequently sent out. By introducing delays and reordering, this attack is hindered.

Unlike the type-0 remailers, the type-I remailers require technical sophistication to use. Users have to either manually construct all of the encrypted parts of a message before sending it, or install a tool like premail [34] that handles the message construction automatically.

## 2.3 Type-II remailers

Although the type-I remailers were, privacy-wise, a great improvement over the type-0 system, they were still vulnerable to *size correlation attacks* or *replay attacks*. In a size correlation attack, an adversary tries to match the messages sent by a given remailer to the messages it received by matching the sizes of the messages. In a replay attack, the adversary makes a copy of one of the messages received by the remailer, and sends many copies of it to that same remailer. The adversary then observes which outgoing message from that remailer gets repeated many times.

*Type-II* or *Mixmaster remailers* were deployed to address these problems. [41] Type-II remailers divide all messages into a number of fixed-sized packets that are sent separately through the network of remailers in order to defeat size correlations. These remailers also employ more complex techniques to defeat replay attacks.

Messages for type-II remailers can not be constructed manually in any reasonable way; users need specially customized software in order to send anonymous mail.

## 2.4 Type-III remailers

Type-II remailers were the current state of the art in 2002. What has happened in the last five years? A design for *type-III* or *Mixminion remailers* has been proposed [13], which improves privacy protection in a number of ways. First, type-III remailers provide a better system for handling replies to anonymous messages. Type-II remailers only support anonymity—not pseudonymity. In order to receive replies to a type-II message, senders have to set up a pseudonym with the older type-I remailer network.

Type-III remailers also provide improved protection against replay attacks and against *key compromise attacks*, where an attacker learns the private decryption key of one or more of the remailers. The type-III system has several other new features to prevent other forms of attack, and to aid in the management of the network.

Unfortunately, support for type-III remailers is not yet widespread. The implementation of the published design has never been released past the testing stage, and has had almost no work done on it in the last year. Although there are about thirty type-III remailers scattered around the world (about the same as the number of type-II remailers), the authors of Mixminion specifically warn users that "you shouldn't trust Mixminion with your anonymity yet" [14].

# 3 Interactive anonymity and pseudonymity systems

Today's online communication is increasingly interactive and real-time, using technologies like instant messaging. Protecting these types of communication, as well as other interactive Internet applications, such as the world-wide web, remote logins, voice-over-IP and games, poses a much more significant challenge than the corresponding problem for email. Whereas remailers obtain much of their security from delaying and reordering messages, such delays are unacceptable in the context of low-latency interactive services, and tradeoffs often have to be made.

In 1995, Wei Dai presented a design of an anonymity system for low-latency traffic, which he called "PipeNet" [12]. The design of PipeNet emphasized security over all else: if the system detected any anomaly that *could* be an attacker trying to compromise privacy, the entire network would shut itself down. Of course, no realistic system could work this way; people simply wouldn't use it. There have been a number of systems that have been implemented and fielded over the years to provide practical security and privacy to users of interactive Internet applications. We examine several of these next.

## 3.1 Anonymizer.com

Anonymizer.com, a company we mentioned in the 2002 survey, continues to run the Anonymizer proxy service, a system we first mentioned in the 1997 survery. [1] They continue to be one of the few commercially successful anonymity technology providers. The Anonymizer works much like the type-0 remailers: a web browser makes a request to the Anonymizer, which relays the request to the intended web server. This service protects the user's privacy from that web server, but not from Anonymizer.com itself, or from anyone watching the Internet near it. As we saw in 2002, by providing protection only against this simpler threat model, Anonymizer.com is able to keep costs and complexity down.

## 3.2 Onion Routing

The US Naval Research Lab's Onion Routing project [45, 28] was the first PipeNet-like system to be widely deployed. Although its primary use was for anonymizing web traffic, it also allowed users to anonymously connect to any TCP/IP server on the Internet. A user configures his Internet applications to use the SOCKS proxy protocol [33] to connect to an *Onion Proxy*. Analogously to remailer systems, the Onion Proxy creates a path through several *Onion Routers* situated around the Internet.

Unlike remailer systems, however, this path is long-lived. Once it is created, any data sent through this path is anonymously delivered to the intended TCP/IP server. Any replies from that server are returned along the path to the Onion Proxy, and from there to the user's application. When the application is finished communicating with the server, the path is torn down, freeing the resources allocated for it at the Onion Routers.

The original deployed Onion Routing network was primarily a proof-of-concept; it later evolved into the Tor network (see below).

## 3.3 The Freedom Network

The Freedom Network was a commercial venture by Zero-Knowledge Systems, Inc. [5] Also a PipeNet-inspired system, it incorporated some of the ideas from the Onion Routing project, but its design differed in important ways. For example, while Onion Routing was a TCP/IP-based system that could anonymously transport any TCP/IP protocol, the Freedom Network was an IP-based system that could transport UDP/IP as well. Unlike Onion Routing's pure anonymity, the Freedom Network provided a persistent pseudonymity service, enabling users to maintain separate online personas. It also used protocol-specific techniques to protect both the users of the network and the network itself. Importantly, Freedom removed the need for users to configure their Internet applications, which removed the potential for privacy-degrading mistakes.

The Freedom Network recruited operators all over the world to run its *AIP nodes* (Anonymous Internet Proxies, again analogous to remailers), and paid them to do so. Unfortunately, as we mentioned in the 2002 survey, these costs proved to be prohibitive; there were not enough paid users to support the high-quality network that a commercial venture requires, and the network had already been shut down by that time.

## 3.4 Java Anon Proxy

Java Anon Proxy (JAP) is a project of Technical University Dresden. [23] It is one of the few privacy-enhancing technologies that was around in 2002, and is still in use today. Unlike PipeNet-based systems, JAP is a web-only anonymization tool that uses the techniques of type-II remailers to do its job. Web requests and replies are divided into fixed-sized chunks, and sent through a series of mix nodes. Each

such node collects a batch of these chunks, encrypts or decrypts them as appropriate, reorders them, and sends them on to the next mix node.

As with Onion Routing, users protect their privacy with JAP by running the JAP client program, and configuring their web browsers to use the JAP client as an HTTP proxy. In this way, each of the user's web requests is sent to the JAP client, which divides it up into chunks, and sends those chunks through the mix network.

### 3.5 Tor

Tor [19, 18] is a new system that has appeared since the 2002 paper. It is the next generation of the Onion Routing project, and it is the most successful (in terms of number of users) interactive anonymity tool to date. Hundreds of thousands of users send about 8 terabytes of traffic per day through hundreds of Tor nodes. As it is an extension of the Onion Routing project, it shares many of that project's characteristics: it only anonymizes TCP/IP protocols, it requires configuration of users' Internet applications, and so on.

Unlike the erstwhile Freedom Network, the Tor nodes are run by volunteers and all of the software is free and open-source. Although somewhat cumbersome for novice users to install and use on its own, graphical user interfaces such as Vidalia [21] and other helpful tools like Torbutton [43] greatly enhance Tor's ease of use.

Currently, one of Tor's biggest drawbacks is its noticeable degradation to web browsing speeds. Ideally, Tor could be used in an "always on" mode, with users not even noticing its presence. Although Tor's sluggish performance prevents this today, work is being done to improve the situation. One possible way to accomplish this is to use peer-to-peer techniques to improve its scalability, as we suggested in 2002. A different project, MorphMix [40], proposed such a design, but not only was it never widely deployed for general use, it was later shown to contain flaws in its privacy protection [46].

In addition to protecting the users of TCP/IP-based Internet services, Tor also contains a facility to protect *providers* of such services. The most common such *hidden services* are web servers; a user runs a web server somewhere in the world which is only accessible through Tor, and Tor protects the identities of both the user and the provider of the service. In this way, Tor provides a *censorship-resistant publishing* service, which has been used by whistleblowers, for example, to distribute information of public importance [37]. Other censorship-resistant publishing services include the Free Haven [17], FreeNet [8], and Publius [48] projects mentioned in 2002. Of those latter three projects, however, only FreeNet is still being developed and used today. The Wikileaks project [51, 50] uses both Tor and FreeNet in order to provide a censorship-resistant repository of leaked documents, which anyone can easily add to.

# 4 Communication privacy systems

When communicating over the Internet, the above technologies can help keep identity information private, possibly from third parties, and possibly also from other parties to the communication. In addition, correspondents may wish to keep the *contents* of the communication private from third parties. The technologies in this section allow you to do this. Note that it is usually the case that these technologies can be combined with those of the previous sections to protect both a user's identity and the contents of his communication.

It is important to note that with these technologies, all parties to the communication need to have the same (or compatible) systems installed. This is not the case with the technologies in the previous sections; those systems protect their users' privacy without requiring the other parties' cooperation.

## 4.1 PGP and compatible systems

Pretty Good Privacy (PGP) [25, 39] has been available in one form or another for over 25 years. Although newer versions have many more features, PGP's fundamental purpose is to encrypt and/or digitally sign email (and to decrypt it and verify the signatures at the other end, of course). PGP has evolved from a command-line-only program to one with a full-featured graphical user interface, and there are a number of compatible implementations, such as GNU Privacy Guard (gpg) [32] and Hushmail [31].

Users install some PGP-compatible software, and use it to encrypt their email messages before sending them. This can be done manually, but some email programs, including Outlook, Eudora, mutt and pine, have incorporated PGP support, greatly improving its ease of use.

## 4.2 SSL and TLS

As the world-wide web turned into a platform for e-commerce in the late 1990s, it became important to protect the contents of web transactions. Netscape invented the Secure Sockets Layer (SSL) protocol, which in later versions was renamed Transport Layer Security (TLS) [24, 16]. Though not without problems, SSL and TLS are the single most widely used privacy-enhancing technology to date. Their success stems from the fact that every major web browser comes with support for these technologies built right in and that their use is largely invisible to the user. That is, no special installation or configuration needs to be done by end users before they can benefit from these technologies. A web browser will automatically encrypt web requests when communicating with an SSL/TLS web server, and the server will automatically encrypt its responses; no user intervention is needed at all. Later, we will come back to this theme when we examine properties of useful security and privacy technologies.

### 4.3 Off-the-Record Messaging

In the last five years, online communication has increasingly moved from email to instant messaging, especially among younger users. [7] First released in 2004, Off-the-Record Messaging (OTR) [4, 3] is a technology to protect the contents of these instant messaging communications. As the name implies, OTR provides instant messaging users with an "off-the-record" conversation. Much like conversing face-to-face, OTR users can communicate privately and can also repudiate any claims as to the content of their conversation.

Fundamentally, OTR allows instant messaging users to communicate in an encrypted and authenticated manner. When a user Alice sends a message to her buddy Bob using OTR, she is assured that only Bob will be able to read it. In turn, Bob is assured that the message came from Alice and has not been modified en route.

Moreover, OTR offers *deniability*. If Bob tells his friend Charlie what Alice sent him, Bob is able to offer no *proof* of that assertion—Charlie just has to trust him. OTR avoids using traditional non-repudiable digital signatures for authentication of messages; if messages from Alice had been digitally signed, Charlie could easily check the signatures for himself. Instead, OTR uses inherently repudiable message authentication codes to assure Bob that the message really came from Alice, but render him unable to prove that fact to anyone else.

In addition, by taking advantage of the fact that instant messaging conversations are interactive, OTR is able to provide *perfect forward secrecy* to its messages. If Bob's computer is lost, is hacked into, gets a virus, or any such thing, and all of his secrets are stolen, any messages Alice had previously sent Bob would remain secret.

Users clearly could not manually encrypt every instant message they send, so the OTR encryption must be handled in an automatic way. There are three ways that users can integrate OTR into their instant messaging. The first is by using a proxy: the user runs an OTR proxy on her computer, and configures her instant messaging client to talk to that proxy instead of talking directly to the instant messaging server. This technique can be used by users of proprietary instant messaging clients like iChat and Trillian in order to obtain OTR functionality. The second method is by using a plugin: many instant messaging clients have the ability to have their functionality extended by third-party plugin modules. There are OTR plugins available for the gaim, Trillian, and Miranda instant messaging clients. The third method is to have OTR functionality built directly in to the user's client. This is of course the best option, since, like SSL/TLS, the user does not have to install or configure anything special in order to gain some benefit from OTR. The popular Adium X instant messaging client for the OS X operating system has OTR built in.

## 5 Other privacy-enhancing technologies

There are many more privacy-enhancing technologies that have been proposed, but are not yet in widespread use. In this section, we look at three particular technologies; we hope to see progress on these

over the next five years.

## 5.1 Private payments

In 2002, we discussed the disappointing lack of adoption of electronic cash. Today, there are still no serious electronic cash services. It is important to fill this gap in the set of available privacy-enhancing technologies. Not only is it undesirable for there to be centralized records of everything one purchases online, but databases of payment records—including credit card numbers—are routinely stolen from merchants and from credit card processing firms. [15] These losses can lead to both credit card fraud and identity theft.

While alternatives to online credit card transactions, such as PayPal [38], are gaining popularity, a true privacy-protecting electronic cash solution remains elusive. Although the last of the patents protecting DigiCash's original electronic cash protocol has recently expired, the patents were not the only barrier to entry for a potential electronic cash provider. As we mentioned in 2002, making a system widely accepted and interoperable with the "real" money system is a difficult task. In fact, PayPal itself may be in the best position to offer true privacy-friendly payments online; it already has the payment infrastructure, it could easily provide an interface between electronic cash and the rest of the financial system and it has a large installed user base. Skype is also considering adding a payment system to its voice-and-chat offering [22], though no information is yet available about privacy properties that system may or may not have.

## 5.2 Private credentials

As we saw in 2002, private credentials [6] are a way to separate *authorization* from *authentication*. They allow users to prove that they are authorized to access a certain service or gain a certain benefit, while revealing no unnecessary personal information, such as their identities. Rather than Alice proving "I am Alice" to some server, and the server checking that Alice is on the approved-access list, Alice instead proves "I am approved to access this server" without revealing who she is. This obviates any personal information about Alice being stored on the server, removing the possibility of that information being disclosed or stolen. Credentica [11] is expected to release a line of privacy-friendly Digital Credential products based on this technology in the near future.

## 5.3 Anti-phishing tools

A *phishing attack* occurs when a user is directed to a malicious website, often via a link in email or chat. The site appears to be a common site, like a bank, eBay, or PayPal, but is really run by an attacker—the *phisher*. The message encourages the user to log in to the site to address an urgent problem with their account; when the user complies, the phisher captures the login name and password. From there the phisher can hijack the account, steal money or mount an identity theft.

There are a number of tools available to help a user determine if he is looking at an authentic website or at a phishing site. These tools often appear as a toolbar in the user's web browser that turns one of three colours: one colour if the tool determines the site is probably genuine, one if it determines the site is probably a phishing site, and one if it cannot make a determination.

The way these tools make these determinations vary. Some, like eBay's Account Guard [20], compare the URL being visited to centrally maintained lists of good and bad sites. Users can suggest sites to be added to either list, and the list maintainers generally manually verify them before adding them. Other tools, like the Cloudmark Anti-Fraud Toolbar [9], use the collective ratings of its users to automatically mark sites as "genuine" or "phishing". Some, like Google's Safe Browsing toolbar [29], use the fact that genuine sites generally have higher Google PageRank than phishing sites. Many tools use combinations of these techniques.

Zhang et al. [52] present an evaluation of ten of these anti-phishing toolbars and find that they "left a lot to be desired". They give some suggestions for further improvements to toolbars like these; we can only hope the state of the art will advance in the next five years.

# 6 Useful security and privacy technologies

Since 2002, we have indeed seen a small amount of progress; there are a handful of new technologies that people are actually using in order to protect their privacy when they use the Internet. In comparison, *research* in privacy-enhancing technologies in the last five years has been booming. New technologies have been proposed in a number of different academic settings, but many do not make it out of the lab. Worse, some do not even make it from design into working code at all. These technologies do not improve people's security and privacy.

What we would like to see more of are security and privacy technologies that make a real difference to real people. We call such systems **useful security and privacy technologies**, and we have identified a number of properties such technologies must have.

**Usability:** It has long been known that many security and privacy technologies are hard to use, or hard to use correctly. Difficult-to-use technologies frustrate users, and can even put them in the unfortunate situation of believing they are being protected when they in fact are not. [49, 42] In order for a technology to be useful, users need to be *able* to use it, and be able to use it properly. In addition, users have to *want* to use it; if a system protects their privacy at the expense of greatly slowing down their Internet experience, for example, users will simply turn it off.

**Deployability:** In order for a technology to be useful, it must be possible for everyday users doing everyday things to obtain it and benefit from it. This means it needs to be compatible with their preferred operating system, their preferred web browser, their preferred instant messaging client, and so on. Ideally, the technology would be built right in so that the user doesn't even need to find and install separate software packages.

**Effectiveness:** Many designed, and even widely deployed, security and privacy technologies contain flaws that can render their ostensible protection moot. For a technology to be useful, it of course has to work and to give the user the benefit it promises. Open design and open implementation can help experts spot problems before too many users are left vulnerable.

**Robustness:** Some technologies will work as advertised, but only so long as thing go "according to plan". But most technology designers' plans overlook the realities of users on the Internet today: their computers contract worms and viruses, they forget their passwords, they get tricked by phishing attacks, they misunderstand (or just "click through") security-critical dialog boxes, and so on. A useful system needs to maintain as much protection as possible in these situations, since they will occur unfortunately often in practice.

In order to close the gap between the number of systems proposed by researchers and the number of systems giving benefit to users, developers of privacy-enhancing technologies should design with these principles in mind.

## 7 Conclusion

The last five years have seen a small increase in the availability of privacy-enhancing technologies for the Internet, including at least one, Tor, which is seeing significant use. This improvement over the previous half-decade is encouraging, but much work remains. We need more technologies that move all the way from design to widespread use and we suggest that the four principles of useful security and privacy technologies—usability, deployability, effectiveness and robustness—may guide us in the right direction.

## References


[1] Anonymizer.com. Anonymizer - Anonymous Proxy, Anonymous Surfing & Anti Spyware. http://www.anonymizer.com/. Accessed 11 January, 2007.

[2] Attrition.org. DLDOS: Data Loss Database—Open Source. http://attrition.org/dataloss/dldos.html. Accessed 11 January, 2007.

[3] Nikita Borisov and Ian Goldberg. Off-the-Record Messaging. http://otr.cypherpunks.ca/. Accessed 10 January, 2007.

[4] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-Record Communication, or, Why Not To Use PGP. In *Proceedings of the Workshop on Privacy in the Electronic Society 2004*, pages 77–84, Washington, DC, October 2004.

[5] Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom Systems 2.0 Architecture. http://osiris.978.org/~brianr/crypto-research/anon/www.freedom.net/products/whitepapers/Freedom_System_2_Architecture.pdf. Accessed 10 January, 2007.

[6] Stefan Brands. *Rethinking Public Key Infrastructures and Digital Certificates—Building in Privacy*. MIT Press, August 2000.

[7] Dan Carnevale. E-Mail is for Old People. *The Chronicle of Higher Education*, 53(7):A27, 6 October 2006.

[8] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability, Lecture Notes in Computer Science 2009*, pages 46–66, Berkeley, CA, July 2000.

[9] Cloudmark, Inc. Cloudmark - Anti Spam and Spam Blocker Solutions. http://www.cloudmark.com/desktop/howitworks/. Accessed 10 January, 2007.

[10] Consumers Union of US, Inc. CR Investigates: Your privacy for sale. *Consumer Reports*, 71(10):41, October 2006.

[11] Credentica. Credentica - Enterprise Solutions For Identity & Access Management. http://www.credentica.com/. Accessed 10 January, 2007.

[12] Wei Dai. PipeNet 1.1. http://www.weidai.com/pipenet.txt. Accessed 11 January, 2007.

[13] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 2–15, Oakland, CA, May 2003.

[14] George Danezis and Nick Mathewson. mixminion - Type III anonymity client. manual page, http://mixminion.net/manpages/mixminion.1.txt. Accessed 11 January, 2007.

[15] Eric Dash and Tom Zeller Jr. MasterCard Says 40 Million Files Put at Risk. *The New York Times*, page A1, 18 June 2005.

[16] Tim Dierks and Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346, http://www.ietf.org/rfc/rfc4346.txt.

[17] Roger Dingledine, Michael J. Freedman, and David Molnar. The Free Haven Project: Distributed Anonymous Storage Service. In *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability, Lecture Notes in Computer Science 2009*, pages 67–95, Berkeley, CA, July 2000.

[18] Roger Dingledine and Nick Mathewson. Tor: anonymity online. http://tor.eff.org/. Accessed 10 January, 2007.

[19] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, San Diego, CA, August 2004.

[20] eBay, Inc. Using eBay Toolbar's Account Guard. http://pages.ebay.com/help/confidence/account-guard.html. Accessed 10 January, 2007.

[21] Matt Edman and Justin Hipple. Vidalia - Home. http://vidalia-project.net/. Accessed 10 January, 2007.

[22] Joris Evers. What threats does Skype face? http://news.com.com/What+threats+does+Skype+face/2008-7350_3-6146092.html, 2 January 2007. Accessed 10 January, 2007.

[23] Hannes Federrath. JAP — Anonymity & Privacy. http://anon.inf.tu-dresden.de/index_en.html. Accessed 10 January, 2007.

[24] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL Protocol—Version 3.0. Internet Draft, http://wp.netscape.com/eng/ssl3/draft302.txt.

[25] Simson Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly, December 1994.

[26] Ian Goldberg. Privacy-enhancing Technologies for the Internet, II: Five Years Later. In *Workshop on Privacy Enhancing Technologies 2002, Lecture Notes in Computer Science 2482*, pages 1–12. Springer-Verlag, April 2002.

[27] Ian Goldberg, David Wagner, and Eric A. Brewer. Privacy Enhancing Technologies for the Internet. In *COMPCON '97*, pages 103–109, February 1997.

[28] David Goldschlag, Michael Reed, and Paul Syverson. Onion Routing for Anonymous and Private Internet Connections. *Communications of the ACM*, 42(2):39–41, 1999.

[29] Google, Inc. Google Safe Browsing for Firefox. http://www.google.com/tools/firefox/safebrowsing/. Accessed 10 January, 2007.

[30] Sabine Helmers. A Brief History of anon.penet.fi—The Legendary Anonymous Remailer. *Computer-Mediated Communication Magazine*, 4(9), September 1997. http://www.december.com/cmc/mag/1997/sep/helmers.html.

[31] Hush Communications Corp. Hushmail - Free Email with Privacy. http://www.hushmail.com/. Accessed 10 January, 2007.

[32] Werner Koch. The GNU Privacy Guard. http://www.gnupg.org/. Accessed 10 January, 2007.

[33] Marcus Leech, Matt Ganis, Ying-Da Lee, Ron Kuris, David Koblas, and LaMont Jones. SOCKS Protocol Version 5. RFC 1928, http://www.ietf.org/rfc/rfc1928.txt.

[34] Raph Levien. Premail. http://www.mirrors.wiretapped.net/security/cryptography/apps/mail/premail/. Accessed 11 January, 2007.

[35] Michael Liedtke. Stolen UC Berkeley laptop exposes personal data of nearly 100,000. *Associated Press*, 28 March 2005.

[36] Leslie Miller. Report: TSA Misled Public on Personal Data. *Associated Press*, 25 March 2005.

[37] MindFreedom International. Eli Lilly Targets Free Speech on MindFreedom's Web Site in Battle Over Zyprexa Documents. http://www.mindfreedom.org/know/psych-drug-corp/eli-lilly-secrets/. Accessed 10 January, 2007.

[38] PayPal. Privacy - PayPal. https://www.paypal.com/cgi-bin/webscr?cmd=xpt/general/Privacy-outside. Accessed 10 January, 2007.

[39] PGP Corporation. PGP Corporation - Products - PGP Desktop Email. http://www.pgp.com/products/desktop_email/index.html. Accessed 10 January, 2007.

[40] Marc Rennhard and Bernhard Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In *Proceedings of the Workshop on Privacy in the Electronic Society 2002*, pages 91–102, Washington, DC, November 2002.

[41] Len Sassaman and Ulf Möller. Mixmaster. http://mixmaster.sourceforge.net/. Accessed 11 January, 2007.

[42] Steve Sheng, Levi Broderick, Colleen Alison Koranda, and Jeremy J. Hyland. Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software. Poster session, 2006 Symposium On Usable Privacy and Security, Pittsburgh, PA, July 2006.

[43] Scott Squires. Torbutton. http://freehaven.net/~squires/torbutton/. Accessed 10 January, 2007.

[44] Marina Strauss and Sinclair Stewart. Computer breach exposes TJX shoppers to fraud; Parent of Winners, HomeSense targeted. *The Globe and Mail*, page B3, 18 January 2007.

[45] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous Connections and Onion Routing. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, CA, May 1997.

[46] Parisa Tabriz and Nikita Borisov. Breaking the Collusion Detection Mechanism of MorphMix. In *Workshop on Privacy Enhancing Technologies 2006, Lecture Notes in Computer Science 4258*, pages 368–383. Springer-Verlag, June 2006.

[47] US Postal Service. Identity Theft. http://www.usps.com/postalinspectors/idthft_ncpw.htm. Accessed 11 January, 2007.

[48] Marc Waldman, Aviel Rubin, and Lorrie Cranor. Publius: A robust, tamper-evident, censorship-resistant web publishing system. In *Proceedings of the 9th USENIX Security Symposium*, pages 59–72, Denver, CO, August 2000.

[49] Alma Whitten and J.D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, Washington, DC, August 1999.

[50] Wikileaks. wikileaks.org. http://www.wikileaks.org/index.html. Accessed 17 January, 2007.

[51] Elizabeth Williamson. Freedom of Information, the Wiki Way; Site to Allow Anonymous Posts of Government Documents. *The Washington Post*, page A13, 15 January 2007.

[52] Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong. Phinding Phish: Evaluating Anti-Phishing Tools. In *Proceedings of the 14th Annual Network & Distributed System Security Symposium (NDSS 2007)*, San Diego, CA, February 2007.