Privacy-enhancing technologies for the Internet, II: Five years later

Ian Goldberg

Zero-Knowledge Systems, Inc.
ian@zeroknowledge.com

Abstract. Five years ago, "Privacy-enhancing technologies for the Internet" [23] examined the state of the then newly emerging privacy-enhancing technologies. In this survey paper, we look back at the last five years to see what has changed, what has stagnated, what has succeeded, what has failed, and why. We also look at current trends with a view towards the future.

1 Introduction

In 1997, the Internet was exploding. The number of people online was more than doubling every year, thanks to the popularity of email and the World Wide Web. But more and more of these people came to realize that anything they say or do online could potentially be logged, archived, and searched. It turns out this was not simply an idle fear; today, the WayBack Machine [30] offers archives of the World Wide Web back to 1996, and Google Groups [26] offers archives of Usenet newsgroups back to 1981! Even in 1996, the cartoon "Doctor Fun" recognized this problem enough to joke, "Suddenly, just as Paul was about to clinch the job interview, he received a visit from the Ghost of Usenet Postings Past." [17].

So-called *privacy-enhancing technologies* were developed in order to provide some protection for these visitors to cyberspace. These technologies aimed to allow users to keep their identities hidden when sending email, posting to newsgroups, browsing the Web, or making payments online.

The 1997 paper "Privacy-enhancing technologies for the Internet" [23] surveyed the landscape of past and then-current privacy-enhancing technologies, as well as discussing some promising future candidates. The need for privacy has not diminished since then; more people are still getting online and are being exposed to privacy risks. Identity theft [48] is becoming a bigger problem, and we are even seeing explicitly privacy-degrading technologies being deployed by companies like Predictive Networks and Microsoft, who aim to track consumers' television-watching habits [13]. The need for privacy is still very real, and technology is our main tool to achieve it.

In this paper, we take a second look around the privacy-enhancing technology landscape. Although there are many such technologies, both in the offline and online worlds, we focus our attention on technologies aimed at protecting Internet users, and even then, we primarily discuss only technologies which have seen some amount of deployment. In our look around, we see some parts that appear just the same as five years ago, often surprisingly so; we see some parts that are new, but not where we expected them.

Finally, we take a stab at future directions, perhaps setting ourselves up for a sequel in another five years.

2 What Was

In this section, we recap the state of the world in 1997. For more detail, the interested reader is referred to the original paper [23].

2.1 What Was Well-Established

In 1997, *anonymous remailers* for electronic mail were established technology. At first, there was the original "strip-headers-and-resend" style of remailer (also known as a "type 0" remailer), the best-known example of which was anon.penet.fi. The penet remailer also allowed for replies to anonymous posts: when you sent your first message through the system, you were assigned a fake address at the anon.penet.fi domain as a pseudonym (or "nym"). The headers on your original email would then get re-written to appear to come from that nym. Replies to that nym would cause the remailer to look up your real email address in a table it kept, and the reply message would be forwarded back to you.

Unfortunately, in 1996, legal pressure forced the operator, Johan Helsingius, to reveal user addresses stored in the table of nyms. In order to prevent further addresses from being forced to be divulged, Helsingius shut down the widely used remailer completely. [28]

To combat the problems of a single operator being able to lift the veil of anonymity, either because he was a bad actor, or because he was able to be coerced, a new style of remailer was developed. These were called "type I" or "cypherpunk-style" remailers [11]. To use type I remailers, a user sends his message not via a single remailer, as with type 0, but rather, selects a *chain* of remailers, and arranges that his message be successively delivered to each remailer in the chain before finally arriving at the mail's inteded destination.

These type I remailers also supported PGP encryption, so that each remailer in the chain could only see the address of the next remailer, and not the ones further down, or that of the final recipient (or even the body of the message). Only the last remailer in the chain (called the "exit node") could see the address of the receipient, and the body of the message.

2.2 What Was Current

Type I remailers had always had some issues with security; for example, an attacker who could watch messages travel through the remailer network could easily trace messages from their destination back to their source if they were not encrypted, and even if they were, he could do the same simply by examining the sizes of the encrypted messages.

In order to fix this and other security problems with the type I remailers, "type II", or "Mixmaster" remailers were developed [45]. Mixmaster remailers always used chaining and encryption, and moreover, broke each message into a number of fixed-size packets, and transmitted each packet separately through the Mixmaster chain. The exit node recombined the pieces, and sent the result to the intended recipient. This added to the security of the system, but at the cost of requiring special software to send email through the Mixmaster network (but no special software was required to receive email). In contrast, a type I message could be constructed "by hand" in a straightforward manner.

Type II remailers provided an excellent mechanism for *sending* email without revealing your identity. The most popular technique for securely arranging to *receive* email was the use of the "newnym" style nymserver [36]. This technology allowed you to pick a pseudonymous address at the nym.alias.net domain, and have that address associated to a "reply block", which is a multiply-encrypted nested chain of addresses, much in the style of a type I remailer message. The remailer network, when presented a message and a reply block, would forward the message along each step in the chain, eventually causing it to reach the owner of the pseudonym.

Another technique for receiving messages was the use of *message pools*. Simply arrange that the message be encrypted and posted to a widely distributed Usenet newsgroup such as alt.anonymous.messages. Since everyone gets a copy of every message, it's not easy to tell who's reading what.

Using a combination of the above techniques, anonymous and pseudonymous email delivery was basically a solved problem, at least from a technical point of view.

Attention turned to other directions; email is not the only interesting technology on the Internet. The next obvious choice was the World Wide Web. In 1997, the state of the art was roughly equivalent to the technology of type 0 remailers for email. The Anonymizer [2] was (and still is) a web proxy you can use to hide your IP address and some other personal information from the web sites you visit. Your web requests go from your machine, to the Anonymizer, to the web server you're interested in. Similarly, the web pages come back to you via the Anonymizer.

Finally, technology was being rolled out in 1997 for the use of anonymous digital cash. The promise of being able to pay for things online in a private fashion was enticing, especially at a time when consumers were still being frightened away from using their credit cards over the Internet. There were a number of companies rolling out online payment technologies; the most privacy-friendly technology involved was invented by David Chaum and was being commercialized by a company called DigiCash [9].

2.3 What Was Coming

The short-term horizon in 1997 had a number of special-purpose projects being studied. Ross Anderson's "Eternity Service" [1] (later implemented in a simpler form by Adam Back as "Usenet Eternity" [4]) promised the ability to publish documents that were uncensorable. In Back's implementation, the distributed nature of Usenet was leveraged to provide the redundancy and resiliancy required to ward off attempts to "unpublish" information.

Perhaps the most promising upcoming technology in 1997 was Wei Dai's proposal for "PipeNet": a service analogous to the remailer network, but designed to provide anonymity protection for *real-time* communication, such as web traffic, interactive chats, and remote login sessions [12]. The additional difficulties imposed by the real-time requirement required significant additions in complexity over the remailer network. However, the range of new functionality potentially available from such a system would be considerable. For example, without a PipeNet-like system, anonymous digital cash isn't very useful: it would be like sending an envelope with cash through the mail, but putting your (real) return address on the envelope.

Unfortunately, PipeNet was never developed past the initial design stages. Onion Routing [24] was another project that was starting to get deployed in 1997, and which was attempting to accomplish similar goals as PipeNet. Onion Routing, however, elected to trade off more towards performance and robustness; in contrast, PipeNet chose security and privacy above all else, to the extent that it preferred to shut down the entire network if the alternative was to leak a bit of private information.

3 What Happened Since

So that was 1997. It's now 2002. What changes have we seen in the privacy-enhancing technology landscape in the last five years?

The widespread consumer acceptance of the World Wide Web has led to further research into privacy protection in that space. An example is Crowds [43], an AT&T project which aims to apply the principles of type I remailers to the World Wide Web. The tag line for the project is "Anonymity Loves Company". The principle is that the set of people utilizing this system forms a *crowd*. A web request made by any member of the crowd is either submitted to the web server in question (as would be the usual case for web surfing without the Crowds system), or else sent to another member of the crowd. (The choice is made randomly.) If it is sent to another member, that member again randomly decides whether to submit the request or to pass it off to another member, and so on. Eventually the request makes it to the web server, and the response is handed off down the chain of requesting members until it reaches the member who originated the request.

The idea is similar to that of chaining used in type I remailers, but with a couple of notable differences. First, unlike in the remailer case, the chain used is *not* selected by the user, but is instead randomly generated at a hop-by-hop level. Also, cryptography is not used to protect the inter-member communications. This reflects the different threat model used by Crowds: it is only trying to provide plausible deniability against the web server logs compiled by the site operator; for example, if it is observed that your machine made a web request for information about AIDS drugs, it is known only that some member of the worldwide crowd requested that information, not that it was you. Crowds makes no attempt to thwart attackers able to sniff packets across the Internet.

A recent German project, JAP (Java Anonymous Proxy), aims to protect against a larger class of threats, though still only with respect to protecting the privacy of people browsing the Web [21]. JAP applies the ideas of type II remailers to web surfing; requests and responses are broken up into constant-size packets, encrypted, and routed through multiple intermediate nodes, called mixes. Each mix waits for a number of packets to arrive, then decrypts one layer from each, and sends them on their way in a single randomly-ordered batch.

Privacy when *browsing* content on the Web is not the only important consideration; some information is important to distribute, yet may get the distributors in trouble with local authorities or censors. That some information is deemed by a local government somewhere in the world as unsuitable should not mean the provider should be forced to remove it entirely. Server-protecting privacy systems allow for the publishing of information online without being forced to reveal the provider's identity or even IP address.

It should be noted that it is usually insufficient to simply put the information on a public web hosting service, because the provider of that service will simply remove the offending material upon request; it has very little incentive not to comply.

The aforementioned Eternity service was a first attempt to solve this problem. More recently, projects such as Free Haven [14], FreeNet [10], and Publius [49] aimed for similar goals. With Publius, documents are encrypted and replicated across many servers. The decryption keys are split using a secret-sharing scheme [46] and distributed to the servers. A special URL is constructed that contains enough information to retrieve the encrypted document, find the shares of the key, reconstruct the decryption key, and decrypt the document.

Publius cryptographically protects documents from modification, and the distributed nature attempts to ensure long-term availability. In addition, the encrypted nature of the documents provides for deniability, making it less likely that the operators of the Publius servers would be held responsible for providing information they have no way to read.

With Free Haven, the aim is also to provide content in such a manner that adversaries would find it difficult to remove. Free Haven also provided better anonymity features to publishers than did Publius.

More ambitiously, a couple of projects aimed to implement systems somewhat along the lines of PipeNet. As mentioned above, the Naval Research Lab's Onion Routing [24, 25] provided (for a while) more general anonymity and pseudonymity services, for applications other than simply web browsing; services such as remote logins and interactive chat were also supported. IP packets were forwarded between nodes situated around the Internet.

A little later, Zero-Knowledge System's Freedom Network [6] rolled out another PipeNet-inspired project, this one as a commercial venture. Eventually, it was the large infrastructure requirement that was to be the Freedom Network's downfall. Whereas the nodes in the remailer network are all run by volunteers, the Freedom Network was a commercial venture, and there were non-trivial costs associated with operating (or paying people or organizations to operate) the nodes. There were costs associated with network management and nym management. In addition, some defenses against traffic analysis (such as link padding) use an exorbitant amount of bandwidth, which is particularly expensive in some parts of the world. Finally, if users are paying for a service, they expect high-quality performance and availability, which are expensive to provide, and were not required in the free, volunteer remailer network.

Zero-Knowledge Systems simply could not attract enough of a paying customer base to support the overhead costs of running a high-quality network. And they were not the only ones; other commercial ventures which operated large-scale infrastructure, such as SafeWeb [44], suffered the same fate.

The lackluster acceptance of electronic cash could be attributed to similar causes. In the last five years, we have seen many protocols for online and offline electronic payment systems, with varying privacy properties (for example, [9],[7],[40]). The protocols are there, and have been around for some time. But no company has successfully deployed it to date. Why is that? For one thing, electronic cash is really only useful when it is widely accepted. Furthermore, in order for it to interoperate with the “real” money system, financial institutions need to be involved. This can have enormous infrastructure costs, which will be challenging to recoup.

Note that one could construct a “closed” ecash-like system, where there is no exchange of value between the ecash system and the rest of the world, which does not have this problem. A slightly more general technology is called “private credentials” [7], in which the holder of a credential can prove quite complicated assertions about his credential without revealing extra information. For example, you could prove that you were either over 65 or disabled (and thus entitled to some benefit), without even revealing which of the two was the case, and certainly without revealing other indentifying information such as your name. Electronic cash can be seen to simply be a special case of this technology, wherein the credential says “this credential is worth $1 and may only be used once”.

Private credentials are highly applicable to the realm of *authorization*, which is important to distinguish from *authentication*. With an authentication system, you prove your identity to some entity, which then looks you up in some table (for example, an access control list), and decides whether you’re allowed to access whatever service. On the other hand, with an authorization scheme, you simply directly prove that you are authorized to access the service, and never reveal your identity. Authorization schemes allow for much more privacy-friendly mechanisms for solving a variety of problems. Juggling many such authorizations, however, can lead to a non-trivial trust management problem. Systems such as KeyNote [5] allowed one to make decisions based on authorizations for keys, as opposed to authentication of people.

So far, almost every commercial privacy technology venture has failed, with Anonymizer.com [2] being a notable exception. Originally hosting the Anonymizer (see above), Anonymizer.com also offers services including email and newsgroup access, as well as dial-up Internet access. Compared to other infrastructure-heavy attempts, Anonymizer.com has a relatively simple architecture, at the expense of protecting against a weaker threat model. But it seems that that weaker threat model is sufficient for most consumers, and we are starting to see other companies similarly relaxing their threat models [51].

Why is deploying privacy-enhancing technologies so difficult? One large problem is that, generally, these technologies are not simply software products that an end user can download and run, and in so doing, gain some immediate privacy benefit. Rather, there is often some infrastructure needed to support aggregation of users into anonymity groups; not only does this add to the cost of deployment, but users in this case only really accrue privacy benefits once a large number of them have bought into the system.

We can divide privacy-enhancing technologies into four broad categories, roughly in increasing order of difficulty of deployment:

Single party: These are products, such as spam and ad blockers, and enterprise privacy management systems, that can in fact be installed and run by a single party, and do not rely on some external service, or other users of the system, in order to be effective.

Centralized intermediary: These technologies are run as intermediary services. An intermediary maintains a server (usually a proxy of some sort) that, for example, aggregates client requests. Deploying and maintaining such a server is relatively easy, but if it goes away, the customers lose their privacy advantage. The Anonymizer and anon.penet.fi are examples of technologies in this category.

Distributed intermediary: The technologies in this category, such as the remailer network, Crowds, and the Freedom Network, rely on the cooperation of many distinct intermediaries. They can be made more robust in the face of the failure of any one intermediary, but the cost involved to coordinate and/or incentivize the intermediaries to cooperate may be quite large.

Server support required: This last category contains technologies that require the cooperation of not just a single or a handful of intermediaries, but rather that of every server with which the user wishes to perform a private transaction. An example of a technology in this class is private electronic cash, where every shop at which a user hopes to spend his ecash needs to be set up in advance with the ability to accept it.

In general, technologies whose usefulness relies on the involvement of greater numbers of entities, especially when non-trivial infrastructure costs are involved, will be more difficult to deploy.

4 What May Be Coming

4.1 Peer-to-peer Networks and Reputation

How do we address this problem of deploying expensive infrastructure? The remailer network does it with volunteers; can we expand on that idea? Perhaps we can take a page from the peer-to-peer (p2p) playbook. If a good amount of the infrastructure in the system can be provided by the users of the system themselves (as in the Crowds project, for example), we reduce not only the cost to the organization providing the service, but, in the extreme case, the entire reliance on the existence of the organization itself, making the end users supply the pieces of the infrastucture. A p2p technology builds right in the idea of distributing trust instead of centralizing it. By removing any central target, it provides more resistance against censorship or "unpublishing" attacks.

Peer-to-peer systems are a natural place to put privacy-enhancing technologies for another reason, as well: the most common use of p2p networks today is for file sharing. As was seen in the case of Napster [3], although users really enjoy sharing music and other files over the Internet, most p2p protocols do not have any sort of privacy built into them. Users sharing particular files can, and have been, tracked or identified. Adding privacy technology to a p2p network provides obvious advantage to the user, as well as providing a useful service.

Another problem with today's p2p networks is that anyone can respond to a request incorrectly. There exist programs for the Gnutella network [22], for example, that will respond to any request with a file of your choice (probably advertising). As p2p networks grow, combatting this problem will become important. One solution interacts well with privacy-enhancing technologies; that is the use of *reputation*. A collaborative reputation calculation can suggest the trustworthiness of a user, whether that user is completely identified or pseudonymous.

We are starting to see reputation systems deployed today, in such communities as Ebay [15], Slashdot [39], and Advogato [35]. As more research is done in this area, combining this work with privacy-enhanced peer-to-peer networks, in a manner such as begun by Free Haven [14], is a natural step.

4.2 Privacy of Identity vs. Privacy of PII

Most privacy-enhancing technologies to date have been concerned with privacy of identity; that is, the controlling of the distribution of information about *who* you are. But there are many other kinds of information about yourself that you might want to control. Personally identifiable information, or PII, is any information that could be used to give a hint about your identity, from your credit card number, to your ZIP code, to your favourite brand of turkey sausage.

Consumers are starting to get concerned about the amount of PII that is collected about them, and are looking for ways to maintain some control over it. A number of technologies allow the management of web-based advertisements or HTTP cookies [33], for example. Technologies such as Junkbuster [32] and P3P [42] allow the user to control what ads they see and what cookies they store. P3P even allows the choice to be made based on the website's stated privacy practices, such as whether the user is able to opt out of the PII collection. Private credential technologies such as Brands' [7] allow the user to prove things about himself without revealing extra personal information.

Sometimes, however, it is not an option to prevent the *collection* of the information; some kinds of PII are required in order to deliver the service you want. For example, online retailers need your delivery address and payment information; health care providers need your medical history; people paying you money need your SSN.[1] In response, a number of industry players, for example [29, 37, 47, 50], are rolling out products that:

- help consumers manage to whom they give access to their personal information, and
- help organizations that collect said information keep control over it and manage it according to their stated privacy policies.

This "enterprise-based privacy" aims to provide technology for protecting data that has already been collected, as opposed to preventing the collection in the first place.

However, whereas the consumer obviously has an interest in keeping his personal information private, what incentive does an organization have to do the same? In addition to better customer relationships, organizations which collect personal data today

[1] Sometimes, "need" is a strong word. Although, for example, there are ways to make payments online and arrange deliveries without using your credit card number or physical address, it's unlikely the company you're dealing with will go through the trouble of setting up support for such a thing.

often have to comply with various sorts of privacy legislation, which we will discuss next.

4.3 Technology vs. Legislation

In recent years, we have seen an escalating trend in various jurisdictions to codify privacy rules into local law. Laws such as PIPEDA [41] in Canada, COPPA, HIPAA, and the GLB Act [19, 27, 20] in the US, and the Data Protection Directive [16] in the EU aim to allow organizations misusing personal data to be penalized. The German Teleservices Data Protection Act [8] even *requires* providers to offer anonymous and pseudonymous use and payment services, and prohibits user profiles unless they are pseudonymous.

This is an interesting development, since many in the technology community have long said that the security of one's transactions should be protected by technology, and not by legislation. For example, technologists have often criticized the cellphone industry for spending money lobbying the government to make scanning cellphone frequencies and cloning phones *illegal* rather than implementing encryption that would render it *difficult*, if not impossible. While from a financial point of view, the cellphone companies clearly made the correct decision, the result is that criminals who don't care that they're breaking an additional law still listen in on phone calls, and sell cloned cellphones, and people's conversations and phone bills are not in any way more secure.

What has changed? Why are we now embracing legislation, sometimes without technology to back it up at all?

The reason lies in the differing natures of security and privacy. In a privacy-related situation, you generally have a pre-established business relationship with some organization with whom you share your personal data. An organization wishing to misuse that data is discouraged by the stick of Law.

On the other hand, in a security-related situation, some random eavedropper is plucking your personal information off of the airwaves or the Internet. You usually don't *have* someone to sue or to charge. You really need to prevent them from getting the data in the first place, likely through technological means. In the privacy case, you don't *want* to prevent your health care provider from getting access to your medical history; you just don't want them to *share* that information with others, and as we know from the world of online filesharing [38, 22, 18], using technology to prevent people from sharing data they have access to is a non-trivial problem.[2]

With traditional privacy-enhancing technologies, the onus was entirely on the user to use whatever technology was available in order to protect himself. Today, there are other parties which need to be involved in this protection, since they store some of your sensitive information. Legislation, as well as other social constructs, such as contracts, help ensure that these other parties live up to their roles.

So with or without technology to back it up, lesigslation really is more useful in the privacy arena than in the security field. Of course, it never hurts to have both; for

[2] Some people have (sometimes half-jokingly) suggested that Digital Rights Management [31] techniques from the online music arena could be flipped on their heads to help us out here; a consumer would protect his personal data using a DRM technique, so that it could be used only in the ways he permits, and could not be passed from his health care provider to his health food salesman.

example, the enterprise-based technologies mentioned above can be of great assistance in ensuring compliance with, and enforcement of, relevant legislation. In particular, now more than ever, technologists need to remain aware of the interplay between their technology and the changing legislative environment. [34]

5 Conclusion

The last five years have been hard for privacy-enhancing technologies. We have seen several technologies come and go, and have witnessed the difficulty of deploying systems that rely on widespread infrastructure. The technical tools that remain at our disposal are somewhat weak, and we are unable to achieve bulletproof technological protection.

Luckily, many applications do not require such strength from technology; many applications of privacy are social problems, and not technical ones, and can be addressed by social means. When you need to share your health care information with your insurance provider, you cannot use technology to prevent it from distributing that information; social constructs such as contracts and legislation really can help out in situations like those.

In closing, what strikes us most about the changes in privacy-enhancing technologies over the last five years, is that very little technological change has occurred at all, especially in the ways we expected. Instead, what we see is an increased use of combinations of social and technological constructs. These combinations recognize the fact that the desired end result is not in fact the technological issue of keeping information hidden, but rather the social goal of improving our lives.

References

[1] Ross Anderson. The eternity service. In *Proc. Pragocrypt '96*, pages 242–252, 1996.

[2] Anonymizer.com. Online privacy services. http://www.anonymizer.com/.

[3] Associated Press. Napster settles with Metallica. *Wired News*, 12 July 2001.

[4] Adam Back. The eternity service. *Phrack Magazine*, 7(51), 1 September 1997.

[5] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. KeyNote: Trust Management for Public-Key Infrastructures (Position Paper). In *Security Protocols Workshop*, pages 59–63, 1998.

[6] Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom Systems 2.0 Architecture. http://www.freedom.net/products/whitepapers/Freedom_System_2_Architecture.pdf, December 2000.

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

[8] Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie. Act on the Protection of Personal Data Used in Teleservices (Teleservices Data Protection Act — Teledienstedatenschutzgesetz TDDSG). *Federal Law Gazette I*, page 1870, 1997. http://www.datenschutz-berlin.de/recht/de/rv/tk_med/iukdg_en.htm#a2.

[9] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology, Proc. Crypto '82*. Plenum Press, 1983.

[10] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66, 2000.
[11] Computer Cryptology. Apas anonymous remailer use. http://www.faqs.org/faqs/privacy/anon-server/faq/use/part3/section-3.html, 2 December 2001.
[12] Wei Dai. Pipenet. http://www.eskimo.com/˜weidai/pipenet.txt, February 1995. Post to the cypherpunks mailing list.
[13] Michelle Delio. MS TV: It'll Be Watching You. *Wired News*, 11 December 2001.
[14] Roger Dingledine, Michael J. Freedman, and David Molnar. The Free Haven Project: Distributed Anonymous Storage Service. In *Workshop on Design Issues in Anonymity and Unobservability*, pages 67–95, 2000.
[15] eBay, Inc. The world's online marketplace. http://www.ebay.com/.
[16] European Communities. Data Protection. http://europa.eu.int/comm/internal_market/en/dataprot/law/.
[17] David Farley. Doctor fun. http://www.ibiblio.org/Dave/Dr-Fun/df9601/df960124.jpg, 24 January 1996.
[18] FastTrack. P2P Technology. http://www.fasttrack.nu/.
[19] Federal Trade Commission. Children's Online Privacy Protection Act of 1998. http://www.ftc.gov/opa/1999/9910/childfinal.htm.
[20] Federal Trade Commission. Gramm-Leach-Bliley Act. http://www.ftc.gov/privacy/glbact/.
[21] Hannes Federrath. JAP — Anonymity & Privacy. http://anon.inf.tu-dresden.de/index_en.html.
[22] Gnutella News. Welcome to Gnutella. http://gnutella.wego.com/.
[23] Ian Goldberg, David Wagner, and Eric Brewer. Privacy-enhancing technologies for the Internet. In *Proceedings of IEEE COMPCON '97*, 1997.
[24] David Goldshlag, Michael Reed, and Paul Syverson. Hiding routing information. In *Information Hiding, First International Workshop*, pages 137–150, May 1996.
[25] David Goldshlag, Michael Reed, and Paul Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2), February 1999.
[26] Google. Google groups. http://groups.google.com/.
[27] Health Care Financing Administration. HIPAA Welcome Page. http://www.hcfa.gov/medicaid/hipaa/default.asp.
[28] Johan Helsingius. Johan helsingius closes his internet remailer. http://www.cyberpass.net/security/penet.press-release.html, 30 August 1996.
[29] IBM Corporation. Tivoli Security Products. http://www.tivoli.com/products/solutions/security/products.html.
[30] The Internet Archive. Wayback machine. http://web.archive.org/.
[31] InterTrust Technologies Corporation. InterTrust Strategic Technologies and Architectural Research Laboratory. http://www.star-lab.com/.
[32] Junkbusters Corporation. Internet Junkbuster Headlines. http://internet.junkbuster.com/ijb.html.
[33] David Kristol and Lou Montulli. *HTTP State Management Mechanism*, October 2000. RFC 2965.
[34] Lawrence Lessig. *Code and Other Laws of Cyberspace*. Basic Books, 1999.
[35] Raph Levien. Advogato's trust metric. http://www.advogato.org/trust-metric.html.
[36] David Mazières and M. Frans Kaashoek. The design, implementation and operation of an email pseudonym server. In *Proc. 5th ACM Conference on Computer and Communications Security*, pages 27–36. ACM Press, November 1998.

[37] Microsoft Corporation. Microsoft .NET. http://microsoft.com/net/whatis.asp.
[38] Napster, Inc. Napster. http://www.napster.com/.
[39] Open Source Development Network. Slashdot: News for nerds, stuff that matters. http://slashdot.org/.
[40] PayPal. http://www.paypal.com/.
[41] Privacy Commissioner of Canada. The Personal Information Protection and Electronic Documents Act. http://www.privcom.gc.ca/legislation/02_06_01_e.asp.
[42] Joseph Reagle and Lorrie Faith Cranor. The Platform for Privacy Preferences. *Communications of the ACM*, 42(2):48–55, February 1999.
[43] Michael Reiter and Avi Rubin. Anonymous web transactions with crowds. *Communications of the ACM*, 42(2):32–48, February 1999.
[44] SafeWeb. Safeweb startpage. https://www.safeweb.com/.
[45] Len Sassaman and Ulf Möller. Mixmaster. http://sourceforge.net/projects/mixmaster/.
[46] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.
[47] Sun Microsystems, Inc. Liberty Alliance. http://www.projectliberty.org/.
[48] U.S. Department of Justice. Identity theft and fraud. http://www.usdoj.gov/criminal/fraud/idtheft.html, 5 June 2000.
[49] Marc Waldman, Avi Rubin, and Lorrie Faith Cranor. Publius: a robust, tamper-evident, censorship-resistant and source-anonymous web publishing system. In *Proc. 9th Usenix Security Symposium*, pages 59–72, August 2000.
[50] Zero-Knowledge Systems, Inc. Enterprise Solutions. http://www.zeroknowledge.com/business/.
[51] Zero-Knowledge Systems, Inc. Freedom WebSecure. http://www.freedom.net/products/websecure/.