

Dartmouth College

## Dartmouth Digital Commons

---

Computer Science Technical Reports

Computer Science

---

7-29-2003

### Using caching for browsing anonymity

Anna M. Shubina  
*Dartmouth College*

Sean W. Smith  
*Dartmouth College*

Follow this and additional works at: [https://digitalcommons.dartmouth.edu/cs\\_tr](https://digitalcommons.dartmouth.edu/cs_tr)



Part of the [Computer Sciences Commons](#)

---

#### Dartmouth Digital Commons Citation

Shubina, Anna M. and Smith, Sean W., "Using caching for browsing anonymity" (2003). Computer Science Technical Report TR2003-470. [https://digitalcommons.dartmouth.edu/cs\\_tr/220](https://digitalcommons.dartmouth.edu/cs_tr/220)

This Technical Report is brought to you for free and open access by the Computer Science at Dartmouth Digital Commons. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of Dartmouth Digital Commons. For more information, please contact [dartmouthdigitalcommons@groups.dartmouth.edu](mailto:dartmouthdigitalcommons@groups.dartmouth.edu).

# Using caching for browsing anonymity

ANNA M. SHUBINA  
Dartmouth College

SEAN W. SMITH  
Dartmouth College

July 29, 2003

**Dartmouth Computer Science Technical Report TR2003-470**

## **Abstract**

Privacy-providing tools, including tools that provide anonymity, are gaining popularity in the modern world. Among the goals of their users is avoiding tracking and profiling. While some businesses are unhappy with the growth of privacy-enhancing technologies, others can use lack of information about their users to avoid unnecessary liability and even possible harassment by parties with contrary business interests, and to gain a competitive market edge.

Currently, users interested in anonymous browsing have the choice only between single-hop proxies and the few more complex systems that are available. These still leave the user vulnerable to long-term intersection attacks.

In this paper, we propose a caching proxy system for allowing users to retrieve data from the World-Wide Web in a way that would provide recipient unobservability by a third party and sender unobservability by the recipient and thus dispose with intersection attacks, and report on the prototype we built using Google.

## **1 Introduction**

Following the most recent political and economic developments, from increased consumer profiling for many ends and purposes (see, for example, [Ele03]) to plans of the US Government for Terrorism Information Awareness (formerly Total Information Awareness, see [DAR03]), a number of previously unconcerned people started to contemplate being more careful with exposing or giving out any information about themselves. A number of businesses that have been collecting data about their customers have also taken damage from adversary parties. It is no longer a purely hypothetical possibility that a vendor of electronic equipment will find his web logs subpoenaed on murky legal grounds. An example is the recent DirecTV case, where customer lists of a number of different companies that sold smart cards were seized, and people on these lists accused of piracy [Pou03].

The lack of privacy of transactions on the web being well-known (as documented, for example, at [Ele97]), more people are now starting to try services that claim to provide anonymous browsing on the web, such as `anonymizer.com` or `the-cloak.com`.

Merely doing a Google search provides one with a long list of what seems like a number of very different services - free or paid, with more or less features and paranoia, with better or worse performance, with bigger or smaller quotas - which claim that they would anonymize one's web browsing. Looking closer, however, one cannot help noticing that most of them are based on the same underlying assumptions, the same attacker model and the same design.

The few more complicated systems that consider a more powerful attacker have yet to go into wide use and require extra software download and setup from the user (and some of them may also come with extra obligations, such as automated request-forwarding). Of these, we are aware of only two that are currently publicly available: Crowds [RR98] and JAP [BFK01]. The first commercial mix-net, Freedom Network [GS01] is now offline permanently [SG00]; Onion Routing [SGR97] is offline temporarily (we hope), waiting for the test of its second generation system.

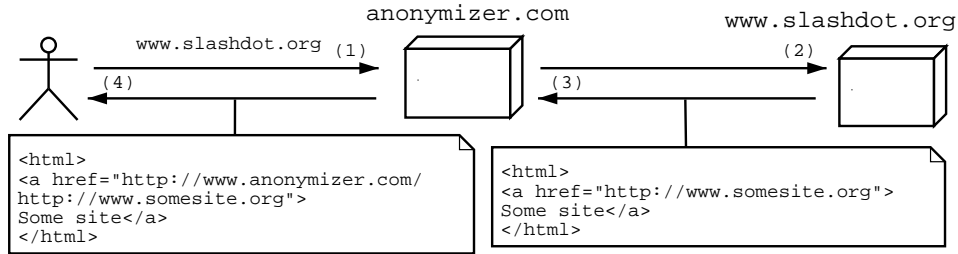
This does not leave many options to a user concerned about his privacy, raising the question: what are the simplest anonymizing systems really giving a user in search of protection, and is there a way to get better protection than that?

## 2 Anonymity and Unobservability

In the terms of [PK00], *anonymity* is the state of not being identifiable within a set of subjects. *Sender anonymity* means that the sender of the message cannot be identified, while *recipient anonymity* means that the recipient of the message cannot be identified. *Unlinkability of sender and recipient* means that it cannot be identified that a given sender and recipient are communicating with each other. Even stronger than anonymity, *unobservability* means that the existence of the message itself cannot be detected. *Sender (or recipient) unobservability* means that it is impossible to detect whether any sender (recipient) from the set of subjects is sending (receiving), while *relationship unobservability* means that it is impossible to detect the existence of a message sent from a possible sender to a possible recipient.

For this definition to work, one needs to consider also two concepts: that of *the attacker* against whom anonymity is achieved and of *the degree of anonymity* (as described in [RR98]), the certainty with which the attacker can pinpoint the sender, the recipient, or link them to each other. The attackers can be *passive* - merely watching the traffic, or *active* - capable of inserting their own traffic. They can be *local*, watching only one node or wire, or *global*, watching multiple nodes or wires. They can watch traffic over short or long periods of time. Theoretical treatments also need to consider the computational power of the attacker.

Figure 1: **Single-hop proxy**



### 3 Anonymizing services on the World-Wide Web

**Single-hop proxy functionality.** Most anonymizing services currently available to the general public at their base are merely single-hop proxies that make the http request for the user. While using a single-hop proxy, the user submits the destination URL to the service and the service immediately issues an http request to this URL. The http request appears to have originated not at the user's computer but at the proxy. As the target computer replies, sending back an html document, the proxy sends back to the user this document with all links rewritten so that they point back to the proxy, not to the sites they originally pointed to. This means that when using, for example, the free part of `anonymizer.com`, a link to `http://slashdot.org` gets rewritten, becoming `http://anon.free.anonymizer.com/http://slashdot.org`.

The connection of the user to the proxy may or may not be encrypted. Some services (see, for example, `the-cloak.com`) provide both; some (see, for example, `anonymizer.com`) provide only unencrypted surfing for free and charge for encrypted surfing.

Besides concealing the user's IP and possibly encryption, extra functionality of anonymizing single-hop proxies may include: filtering out or specially handling cookies; filtering out or rewriting JavaScript, Java, or other active content; filtering out advertisements and banners; proxying or blocking https; faking the `http_user_agent` field in the http header (that is, not revealing information about the user's OS and browser); faking the `http_referer` field (that is, not disclosing the previously visited site).

**Single-hop proxy examples.** Table 1 lists a few available services. Note that most such proxies aren't run by businesses that have obligation to keep them up, and therefore may appear and disappear unpredictably. Some of these services (such as, for example, `anonymizer.com`) have been subject to public review for a long time and have a reputation for not violating their users' privacy. For others, caveat emptor may apply.

**Single-hop proxy attacker model.** A single-hop proxy attempts to protect the identity of the sender of a request from the attacker who can monitor the traffic of the destination site.

Name	URL	Encryption
Anonymizer	<a href="http://www.anonymizer.com">http://www.anonymizer.com</a>	paid version only
the-Cloak	<a href="http://www.the-cloak.com">http://www.the-cloak.com</a>	yes
ProxyWeb.net	<a href="http://www.proxyweb.net">http://www.proxyweb.net</a>	paid version only
SnoopBlocker.com	<a href="http://www.snoopblocker.com">http://www.snoopblocker.com</a>	paid version only
Proxify.com	<a href="http://proxify.com">http://proxify.com</a>	no
Anonymouse	<a href="http://anonymouse.ws">http://anonymouse.ws</a>	no
Web Warper	<a href="http://webwarper.net">http://webwarper.net</a>	no
Anonymization	<a href="http://www.anonymization.net">http://www.anonymization.net</a>	no
PurePrivacy	<a href="http://www.pureprivacy.com">http://www.pureprivacy.com</a>	no

Table 1: Some available single-hop proxies as of June 2003

What this really means is, that leaving aside bugs and faults in design or implementation that may make such services able to disclose the information they claim to conceal (an example is in [MS02]) and leaving aside their additional content filtering capabilities, the main function of a single-hop proxy is the concealment of the user’s IP from the site he accesses.

The destination site can do traffic analysis (see, for example, [SK02] or [Ray01]) to learn about and from the user’s browsing patterns. The site can see what paths the user takes inside it (*communication pattern attack*), the intervals of time between requests (*timing attack*), the amount of transmitted data (*packet volume and counting attack*), as discussed in [SK02]. The site may also be able to correlate different accesses by the same user that occur at different times (*intersection attack*).

Single-hop proxies also do not address the model of the attacker who may be watching user’s traffic as it goes from the user to the proxy and from the proxy to the user. Sometimes they do not encrypt such traffic, allowing any observer on its path the full view of what is going on.

Even if the traffic from the user to the proxy and back is encrypted, the global attacker that is able to watch both the user-proxy and the proxy-destination sides of the proxy might be able to see what data the user requests and gets by doing traffic analysis on the ingoing and outgoing traffic.

**Going beyond a single-hop proxy.** In 1981, Chaum introduced the concept of “Mix-nets” (see [Cha81]). “Mix-nets” are groups of servers that provide anonymity by passing user’s traffic through nodes called *Mixes* which may delay, reorder, reencrypt, pad and forward traffic passing through them. In addition to providing sender anonymity, Mix-nets attempt also to provide sender and recipient unlinkability against a global attacker. Among the examples of Mix-nets are Onion Routing [SGR97], Zero Knowledge Systems’ Freedom Network [GS01], Web MIXes [BFK01], Tarzan [FM02].

Among the other approaches to the problem of anonymous web transactions is the Crowds system [RR98], which allows participating nodes to forward requests within their crowd with a certain probability. Crowds attempt to provide sender and recipient anonymity, but unlike the Mix-nets, they do not attempt

to provide sender and recipient unlinkability against a global attacker.

**A problem: long-term intersection attacks.** A subset of intersection attacks and a major unsolved problem in anonymity systems are so-called “long-term intersection attacks” (see [BL02]), where the attacker is able to watch all or many nodes, including a number of entrance and exit points of the users traffic. Users tend to have a certain behaviour while on-line. They tend to send messages to same or similar destinations. If user  $A$  in a mix daily reads a certain site, after very many observations it may be possible to match  $A$  with the outgoing request to that site, using the knowledge of  $A$ ’s presence or  $A$ ’s absence together with the information whether or not the site was visited during that time.

One of the directions for dealing with this is dummy traffic (as discussed in [BL02], [JVZ01]). Dummy traffic consists of messages sent even when the user has nothing to send. A totally different approach is proposed in PipeNet [Dai98], where the whole mix-net is supposed to shut down when one node stops to communicate with the network.

## 4 Using caching for anonymity

We propose an anonymizing web-browsing system that would deal with the global attacker who is able to do long-term observation. The proposal is to make user requests to proxy independent in time from the proxy’s requests to target sites. The proxy will request and cache information on its own and send it to the users when requested. To do this, the proxy server would collect and store documents which its users are likely to request; provide an encrypted channel to access its contents; and rewrite links in the documents provided to the user to point back at the proxy, similarly to the workings of an ordinary single-hop proxy. This would mean that a global observer looking at the traffic to and from the server will see only regularly scheduled cache updates and encrypted user traffic, thus removing correlation between the users requests and the server’s accesses to remote sites.

This caching proxy will act like a single-hop proxy, in that the destination server does not find out who communicated with it except for the proxy. But it does more than the single-hop proxy in many other respects: the destination server does not find out that someone except the caching proxy communicated with it at all, and neither does a third-party observer; the global attacker cannot correlate requests from the sender and to the recipient; and long-term intersection attacks are meaningless since there is no connection between the proxy’s fetches and the user’s requests.

In short, the attacker model includes not only the attacker at the destination site (as that of existing single-hop proxies) but also the global attacker who is able to do not only short-term, but also long-term attacks.

Similarly to the existing single-hop proxies, the user will not be protected from the attacker at the proxy itself.

	Sender anonymity from		Recipient anonymity from		Sender, recipient unlinkability	Single point of failure
	recipient	3rd party	sender	3rd party		
Single-hop proxies	yes	no	no	no	no	yes
Mix-nets	yes	yes	no	yes	short-term	no
Crowds	yes	yes	no	yes	no	no
Unencrypted caching <sup>1</sup>	yes	no	no	no	no	yes
Encrypted caching <sup>1</sup>	yes	yes	no	yes	yes	yes

Table 2: Anonymizing systems compared

## 5 Candidates for a prototype

**Caching proxies that periodically update their content.** The best known example of a caching proxy that collects data on its own for future use is Google’s cache [Goo]. Every few weeks it crawls all the web, collecting, storing and indexing text information from every webpage [BP98]. Attempting to scale to the size of the whole World Wide Web, it has been optimized for memory usage and speed.

Google’s primary objective is indexing text information. It does make the attempt to index images but only caches thumbnail images, relying on the original image being still present on the server. Such usage may be due also to data size considerations, but there are also legal reasons for it. In a recent case, the 9th U.S. Circuit Court of Appeals ruled that it is legal for search engines to cache thumbnail images [Ols03a], but it is yet to be determined whether it is legal to cache a full-size image copy. According to the Court, the use of thumbnail images is fair use, because they ‘do not supplant the need for originals’ and ‘benefit the public by enhancing information-gathering techniques on the Internet’.

Other search engines that allow access to their caches are Comet Web Search [Com] and Gigablast [Gig]. An interesting web cache is Wayback Machine [Way]. This is an internet archive that contains multiple copies of webpages from 1997 to 2002. Similarly to the caches of search engines, it cannot be used for anonymous browsing. Using JavaScript, Wayback Machine handles links to point back to the cache where archived versions are available, but if that fails, it sends the request to the original site.

**“Using Google cache as anonymous surf”.** The idea of using Google cache for anonymity has occurred to Google’s users before, but Google’s design is not targetted towards this goal. Written apparently as a result of such attempts, an FAQ entitled ‘Don’t Use the Google Cache as Anonymous Surf’, [The03] explains that clicking on a link in a cached page would lead you to a page outside Google cache, an embedded image would attempt to load directly from

<sup>1</sup>Proposed for providing anonymity in this paper.

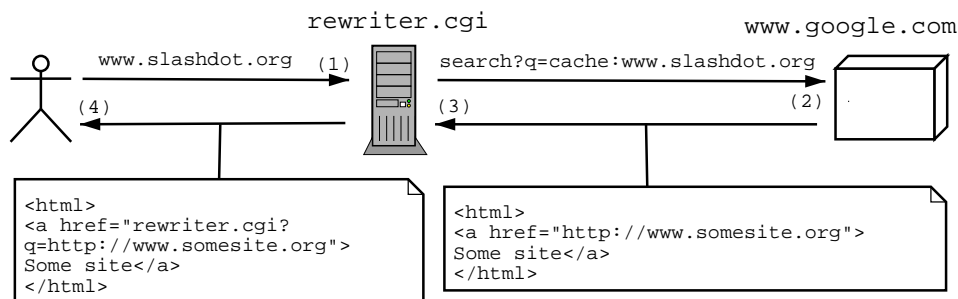


Figure 2: Anonymizing caching proxy

the original site, and redirection might make the whole page load from the original site.

## 6 Prototype

We prototyped<sup>2</sup> an anonymizing caching proxy that outsources all caching to Google by forwarding all requests to Google cache and rewriting links in the resulting documents. When a URL is submitted to the script, the script requests a cached copy from the Google cache. Any link in the document gets rewritten to submit the link to this script, instead of going directly to the requested page. Thus, if we deal with plain text data, every request would be redirected to the Google cache.

The implementation has to detect all outward going links, detect their data type and to rewrite them according to whether their content may be available from Google cache.

Since Google cache does not store images, images are not provided unless the user explicitly specifies that he wants the script to act like a single-hop proxy that fetches them for him from the destination server. Such a transaction would disclose the user's browsing to the global observer. If the script is located at the user's machine, it would also disclose his IP to the destination site, while if the script is not located at the user's machine, this behaviour would be no different from that of a single-hop proxy.

## 7 The caching proxy controversy

It appears to be an unfortunate law of the modern civilization that whenever anyone wants to give out something for free, someone else is going to have a problem with it. Caching proxies are no exception to this rule.

<sup>2</sup>The prototype is at <http://www.cs.dartmouth.edu/~ashubina/google.html>. It is a CGI script that can be located at the user's machine or elsewhere.





EFF is a non-profit group of passionate people &mdash; lawyers, volunteers, and visionaries &mdash; working to protect your digital rights.

[Check us out »](#)

#### HOT TOPICS

Sizzling, current, direct from EFF experts to you.

- » [Anonymity](#)
- » [Anti-Terrorism](#)
- » [Censorship](#)
- » [Copyright Law](#)
- » [Digital Rights Management](#)
- » [Filtering](#)
- » [Surveillance](#)
- » [USA PATRIOT Act](#)

Figure 3: Current [www.eff.org](http://www.eff.org)

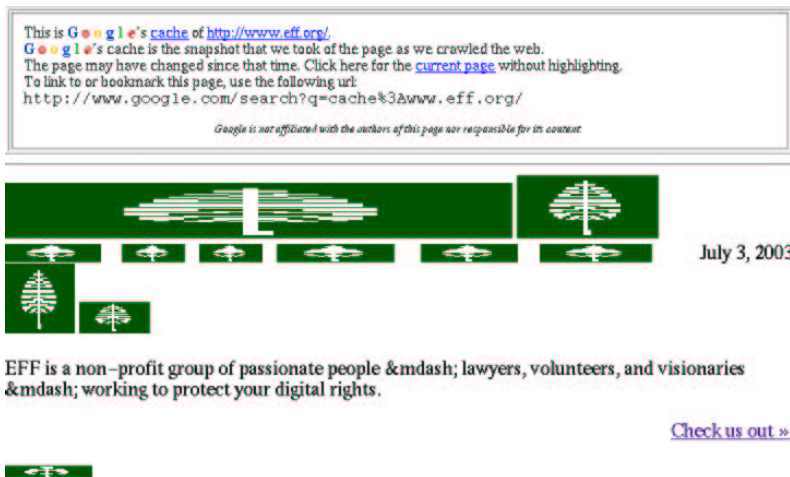


Figure 4: Surfing [www.eff.org](http://www.eff.org) in Google cache (images replaced by the leaf logo of Dartmouth College)

Most Web sites strive for high search engine rankings (on how to play this game, see, for example, [CD03]). However, as described in a recent **news.com** article [Ols03b], many of them are unhappy with caching of their contents.

[Ols03b] gives the following list of complaints about search engine caching:

- **Caching of removed data.** Caching lets users access pages that are temporarily unavailable. Thus it sometimes provides also access to content that has been deliberately taken off-line.
- **Access to registration-only sites.** Due to faults in the design of a number of sites that require registration, caching may end up indexing and caching their content.
- **Detouring traffic from the original sites.** The very feature that we propose to use for anonymity is a subject of such complaints. The claim is that by detouring traffic from the sites where the original information is stored, caching may make Web publishers lose their income.

Interestingly, for the current implementations of all the web caches we reviewed ([Goo], [Way], [Com], [Gig]), the last claim is not quite valid. Among the many ways in which the user's anonymity from the original site provided by the use of such caches is violated are images, which usually include also ads. Since ads are not cached, a cached copy of a webpage with an ad usually displays the ad downloaded from its original provider. Such ads may, however, not be able to track to which site they really owe this request very well, and thus may mistake the web cache for the referrer. (However, the proxy we propose has a different behaviour.)

Google's response to the last complaint is that web sites can easily prevent Google from caching their pages without preventing it from indexing them. All it takes is adding to your page either `<META NAME="ROBOTS" CONTENT="NOARCHIVE">` to exclude all robots, or `<META NAME="GOOGLEBOT" CONTENT="NOARCHIVE">` to exclude just Google. However, some sites are reluctant to do this for fear that this may affect their search rankings.

The Digital Millennium Copyright Act has a narrow exception for Web caching, allowing internet service providers to keep local copies of Web pages. It is not clear, however, whether this would protect also search engine caches or archives similar to [Way] if tested.

The legality of a caching proxy such as we propose under the modern copyright laws is yet to be determined, as is the legality of search engine caches. Due to the very features that make our scheme provide anonymity that are not there for other web caches, our scheme may not be ruled legal even if other web caches are.

## 8 Extensions

Obviously, a proxy such as we propose would not allow the users to use any dynamic content. A user could not provide any information to be passed to the

destination site without informing the destination site and the global observer that someone is using the proxy to look at the destination site. It might be possible to design a proxy that submits such a request for the user together with other people's requests or a number of randomly generated requests, but such an extension is bound to be subject to attacks.

However, a natural extension would be allowing to search Google cache, using Google itself and rewriting the results to point back into the cache. It might also be possible to design a caching proxy that provides thumbnail images, as does Google.

Google (as well as the other caching proxies described above) does not provide encryption, hence the information going to and from Google is not encrypted. An encrypted proxy would allow protection from an attacker that can watch traffic going to it. Against such an attacker, it would also help to bundle packets together or break them up, normalizing their size. Such a proxy could also attempt to conceal its operations from an attacker at the very proxy by means of private information retrieval.

## 9 Conclusion

In this paper we presented an approach that would allow retrieving text information from the web with better anonymity than a single-hop proxy allows to achieve and would also eliminate long-term correlation attacks by the global attacker. The approach uses a caching proxy that contains its own copy of the World Wide Web (or of its large subset). One example of such a proxy is Google cache, which we used for a prototype.

Our prototype is already useful both for achieving limited anonymity, and for accessing cached copies of sites that for some reason are inaccessible at the moment.

As privacy of web transactions becomes more and more important and harder and harder to achieve, it may be worthwhile to use caching proxies to achieve anonymous access to what may be the most important achievement of the human civilization - written text information.

## References

- [BFK01] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. *Lecture Notes in Computer Science*, 2009:115–??, 2001.
- [BL02] Oliver Berthold and Heinrich Langos. Dummy traffic against long term intersection attacks. In *Privacy Enhancing Technologies 2002*. Springer-Verlag, 2002.

- [BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual search engine. In *Seventh International World Wide Web Conference*, 1998.
- [CD03] Tara Calishain and Rael Dornfest. *Google Hacks*. O'Reilly, 2003.
- [Cha81] David Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.
- [Com] Comet. Comet web search. <http://search.cometsystems.com>.
- [Dai98] Wei Dai. PipeNet 1.1. <http://www.eskimo.com/~weidai/pipenet.txt>, 1998.
- [DAR03] DARPA. Report to Congress regarding the Terrorism Information Awareness program. [http://www.darpa.mil/body/tia/tia\\_report\\_page.htm](http://www.darpa.mil/body/tia/tia_report_page.htm), May 2003.
- [Ele97] Electronic Privacy Information Center. Surfer beware: Personal privacy and the internet. <http://www.epic.org/reports/surfer-beware.html>, June 1997.
- [Ele03] Electronic Privacy Information Center. Privacy and consumer profiling. <http://www.epic.org/privacy/profiling>, April 2003.
- [FM02] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, D.C., November 2002.
- [Gig] Gigablast. Gigablast search. <http://gigablast.com>.
- [Goo] Google. Google. <http://www.google.com>.
- [GS01] Ian Goldberg and Adam Shostack. Freedom network 1.0 architecture and protocols, October 2001.
- [JVZ01] Shu Jiang, Nitin H. Vaidya, and Wei Zhao. Power-aware traffic cover mode to prevent traffic analysis in wireless ad hoc networks. In *IEEE Infocom*, 2001.
- [MS02] David Martin and Andrew Schulman. Deanonymizing users of the SafeWeb anonymizing service. Technical Report 2002-003, 11 2002.
- [Ols03a] Stefanie Olsen. Court backs thumbnail image linking. [http://news.com.com/2100-1025\\_3-1023629.html?tag=bplst](http://news.com.com/2100-1025_3-1023629.html?tag=bplst), 2003.
- [Ols03b] Stefanie Olsen. Google cache raises copyright concerns. [http://news.com.com/2100-1032\\_3-1024234.html?tag=fd\\_lede2\\_hed](http://news.com.com/2100-1032_3-1024234.html?tag=fd_lede2_hed), 2003.

- [PK00] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity — a proposal for terminology. In *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability*, volume 2009, pages 1–9. Springer-Verlag, July 2000.
- [Pou03] Kevin Poulsen. DirecTV dragnet snares innocent techies. <http://www.securityfocus.net/news/6402>, July 2003.
- [Ray01] Jean-François Raymond. Traffic analysis: Protocols, attacks, design issues and open problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 10–29. Springer-Verlag, 2001.
- [RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
- [SG00] Adam Shostack and Ian Goldberg. How not to design a privacy system: reflections on the process behind the Freedom product. In *Proceedings of the tenth conference on computers, freedom and privacy: challenging the assumptions*, pages 85–87, Toronto, Ontario, Canada, 2000. ACM Press.
- [SGR97] P F Syverson, D M Goldschlag, and M G Reed. Anonymous connections and Onion Routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, California, 4–7 1997.
- [SK02] Ronggong Song and Larry Korba. Review of network-based approaches for privacy. In *Proceedings of the 14th Annual Canadian Information Technology Security Symposium*, Ottawa, Ontario, May 2002.
- [The03] The Virtual Chase. Don’t use the Google cache as anonymous surf. [http://www.virtualchase.com/ask\\_answer/google\\_cache.html](http://www.virtualchase.com/ask_answer/google_cache.html), May 2003.
- [Way] WayBack. WayBack Machine. <http://www.archive.org>.