Phantom Access Agent: a Client-Side Approach to Personal Information Control

Xiaojian Zhao, Daniel C. Howe, Helen Nissenbaum, David Mazières
NYU Dept. of Computer Science

Abstract

People have criticized on-line services for violating privacy by collecting too much personal information. Though web browsers must generally reveal basic network information such as a user's current IP address, web sites often collect far more, including a user's name, physical location, and email address. Service providers justify their data collection on the grounds that users benefit from such activities as they enable personalization of online experience. Unfortunately, there is no way to evaluate this claim as most services that collect information do so either by default, or as a condition of access, making it difficult or inconvenient for users to avoid revealing personal information. In this paper, we present the Phantom Access Agent (PAA - http://www.scs.cs.nyu.edu/paa/), a lightweight application designed to conceal personal information from online services that require registration as a condition of access. PAA enables users to complete forms with random registration information and facilitates transparent re-registration on subsequent returns with a single button-click. Unlike several other systems that enhance users' choices to share or not share personal information, PAA runs on users' local computers, avoiding dependency on third-parties; whether on the online services themselves to fulfill the promises of their privacy policies or on proxies that offer protection by mediating transactions between individuals and web services. We believe that locating these powers on the client-side better models autonomously chosen privacy preferences. The paper also include a table for quick comparisons among various privacy enhancing technologies, including PAA.

Author Info

Helen Nissenbaum (CONTACT)
Dept. of Culture & Communication
New York University
239 Greene Street, 7th Floor
New York, NY 10003
helen.nissenbaum@nyu.edu
(212)998-5251

Daniel C. Howe
Computer Science Dept.
215 Thompson St. #4
New York, NY 10012
dh254@nyu.edu
(212)777-7159

Xiaojian Zhao
Computer Science Dept.
Warren Weaver Hall, Room 422
251 Mercer Street
New York, NY 10012
xiaojian@cs.nyu.edu
(212)998-3083

David Mazières
Computer Science Dept.
715 Broadway, #708
New York, NY 10003
200203041606.g24G60795304@scs.cs.nyu.edu
(212)998-3492
fax: (212)995-4123

Phantom Access Agent: a Client-Side Approach to Personal Information Control

Xiaojian Zhao, Daniel C. Howe, Helen Nissenbaum, David Mazières
NYU Dept of Computer Science

Abstract
People have criticized on-line services for violating privacy by collecting too much personal information. Though web browsers must generally reveal basic network information such as a user's current IP address, web sites often collect far more, including a user's name, physical location, and email address. Service providers justify their data collection on the grounds that users benefit from such activities as they enable personalization of online experience. Unfortunately, there is no way to evaluate this claim as most services that collect information do so either by default, or as a condition of access, making it difficult or inconvenient for users to avoid revealing personal information. In this paper, we present the Phantom Access Agent (PAA-http://www.scs.cs.nyu.edu/paa/), a lightweight application designed to conceal personal information from online services that require registration as a condition of access. PAA enables users to complete forms with random registration information and facilitates transparent re-registration on subsequent returns with a single button-click. Unlike several other systems that enhance users' choices to share or not share personal information, PAA runs on users' local computers, avoiding dependency on third-parties; whether on the online services themselves to fulfill the promises of their privacy policies or on proxies that offer protection by mediating transactions between individuals and web services. We believe that locating these powers on the client-side better models autonomously chosen privacy preferences. The paper also include a table for quick comparisons among various privacy enhancing technologies, including PAA.

Introduction
Many who oppose new privacy laws for online interactions argue that the restraints imposed by these laws would not only be costly for the gatherers of information – usually commercial and marketing entities – but would also hurt individual users and consumers by denying them many of the benefits of free flowing information such as special offers and, particularly, personalization of online experiences. Opponents of privacy regulation, instead, support an approach known as "self-regulation," promoted as a competitive, free market in privacy, which allows the providers of goods and services to compete with one another by offering varying degrees of privacy and allowing users to choose services accordingly. This way, users who do not object to data collection will opt for the advantages of special offers and highly personalized services while those who do feel strongly about privacy will express their preferences by choosing providers that guarantee it. This approach allows providers to regulate themselves on the dimension of privacy, still offering protections to consumers under existing trade regulations which govern promise breaking and false advertising.

People have both justified and criticized this information gathering on philosophical grounds—weighing users' right to privacy against the rights of publishers to control content. However, those in favor of information gathering often justify their actions with the pragmatic argument that the information they gather allows them to improve their service through personalization. Until now, there has been no good way of evaluating this argument. First, most mainstream online services collect personal information by default without making their practices apparent to users, let alone offering them explicit choices. And even those that maintain a privacy policy (in increasing numbers) place them obscurely on their pages and express their content in proverbial small print, rife with ambiguities, loopholes, and technical jargon. The choice that faces

individuals wishing to take advantage of goods and services offered online is not a rich set of privacy alternatives but, in reality, a "take it or leave it" choice. To truly evaluate claims that users prefer personalization, users would need to be offered a choice between personalization on the one hand, and privacy on the other, for the same service.

One of the reasons this situation exists is that the free market scenario propounded by opponents of privacy regulation relegates users to a passive role; services define the set of options which users are left to take or leave. We believe that a better way to model free choice and properly test the hypothesis that users consistently prefer personalization to privacy is to offer users a richer, more active role in decision-making. This has been the central principle behind our software, Phantom Access Agent (PAA) which offers web users more meaningful control over information exchanges with websites. It functions in cases when users are compelled to register at a website as a precondition of access by allowing users to complete online registration forms anonymously, quickly, and easily. Unlike other privacy protection systems, PAA places almost all of its functionality on users' systems, minimizing dependency on third-parties for the exercise of their choices. In so doing, PAA embodies not only the value of privacy, but also the value of user autonomy.

Work on PAA is part of a larger project on integrating values, like privacy, into the design of web-browser security. According to this project, it is important to consider how system design reflects political and moral values and at the same time to consider important values in the process of design work itself. Unlike other discussions of security that consider privacy to be a contrasting value, we consider privacy an important aspect of security that web browser technology ought to provide individual users. For further discussion on the subject of values in computer system design and other results from the web-browser security project see [40, 41]. For related work, see the initiative on Value Sensitive Design [1]

In the sections that follow we introduce and describe key design and implementation features of PAA. We set a context for PAA by briefly mentioning some of the mechanisms for collecting user information online that have been documented as the source of privacy violations. We also compare and contrast PAA with several other tools that have been developed to protected users against privacy violations. PAA does not "solve" the privacy problem online but tackles merely one aspect of it. So we consider it as part of a larger toolkit of privacy-protecting devices, anticipating potential benefits from use in conjunction with others systems.

1. Background

The most ubiquitous form of 'hidden' data collection in online systems involves the use of 'cookies'. As is documented elsewhere, cookies were initially employed to enable users to transparently return to web sites without having to re-identify themselves. Cookies have since been used in ways that invade users' privacy. For example, the use of cookies by third-party websites to aggregate user profiles to track online activities has been well-documented. The overarching problem can be categorized as involving informed consent: users are neither adequately informed about what cookies do and how personal information is being collected and used, nor adequately given a choice to decline participation[30].

A second technology used to collect user information is the referrer information divulged when downloading images. Such information can be collected through banner ads or even through tiny, transparent images known as 'web-bugs'. Using such techniques, third parties can track users across sites, surreptitiously recording a client's IP address, browser software, and list of pages visited.. When such information is collected, it is possible for a third-party to track a user's browsing behavior over all sites on which they have images. Agencies such as 'DoubleClick'

[31], have used such techniques to aggregate and store such cross-site profiles for users who have never visited their site. Further abuses can occur when such profiles are sold or shared between multiple 3rd party companies and/or linked with users' terrestrial profiles. It has been argued that website privacy policies should be forced to disclose the use of web-bugs and profiling by 3rd parties, however, this topic is rarely mentioned in such documents[42].

A third form of information collection involves sites on which users are obliged to register before accessing content. Registration, often requiring the user to specify a unique username and password, allows service providers to employ a variety of techniques for collecting extensive profiles. An increasing number of sites now require valid email addresses as well to which confirmations are sent on initial registration. This email address can serve as a nearly unique identifier for web site visitors, providing a dangerous opportunity for data aggregation across sites. Such user profile databases have been abused by 'spammers' wishing to send junk mail and sold when businesses have changed hands and privacy policies 'expire'. Additionally, while sound security principles would have users employ different usernames and passwords for each site, many choose the same pair across sites. This username/password combination can be used to violate security at the user's home or work computers if the same password is used there. [23]

Users who prefer not to provide such personal information have a choice of several software utilities that assist in completing registration forms. Buzof [20], Gator [3], Roboform [4], Lucent's Personal Web Assistant (LPWA), and others have been cited as a possible solutions. While the former three assist only in form completion, LPWA generates a number of secure, pseudonymous aliases (personae) for users, including aliased email addresses which are encrypted and stored on a remote LPWA server. . Additionally, LPWA handles coordination between personae and web sites so that users may still receive personalization benefits. LPWA has not, however, addressed the problem of messages routed between users and the remote LPWA server; this has been noted as significant vulnerability in the system[LPWA]. Additionally, in such a scenario, users must trust the identity and 'promised behavior' of third-party individuals and remote systems.

3. Phantom Access Agent

The Phantom Access Agent(PAA) is a light-weight utility designed to conceal personal information from online services requiring registration. It provides auto-completion of registration forms and transparent handling of usernames, passwords, and confirmation mails. Return visits to sites are handled by the inclusion of the PAA Sign-In button on the registration page. A single click generates a new and unique user persona -- including username, email and password -- so that profiling is prevented even within a single site as each visit will appear as a new user. Personal preferences may be maintained within the local SPS database to enable session-level personalization at the site without data re-entry. Unlike other privacy-protecting systems, PAA runs locally on client machines, thereby avoiding dependencies on third parties and remote proxies to mediate transactions between users and web sites.

3.1 Form Completion

PAA consists of two primary modules. The first, the form-completion module is responsible for the auto-completion of registration forms, the inclusion of the PAA Sign-In Button on return visits, and the relaying of messages to and from visited web sites. The second module is the Mail-Responder. The Mail-Responder transparently handles those cases where a registration form becomes valid only upon response to a confirmation email. The following sections will discuss the design and functionality of each module in further detail.

The form-completion module is implemented as a local proxy that relays messages between users and visited sites. While proxies have often been used to defeat user identification schemes, most are located on a remote machine (i.e., a corporate firewall). The form-completion module is different in that it functions transparently on a client machine without requiring special network routing or additional server hardware or software. A local database is maintained of data types and form URLs that has been used for registration on previous visits to each site. When a registration form is visited a second time, the PAA Sign-In Button is inserted into the requested content allowing auto-completion of the form with a single button-click. The form elements and their types are retrieved, formatted and forwarded automatically, with new random substitutions performed on all tagged fields. Fields that have not been tagged are assigned the same value as the previous visit, thus enabling in-session personalization. Additionally, if the user wishes to complete a form by hand, perhaps to gain some feature of cross-session personalization, the Sign-In Button can simply be ignored.

3.2 Mail-Responder

As registration forms often require valid email addresses for the purpose of confirmation, random data submission is not always sufficient. In these cases, the email address is usually verified and a response sent that either requires a reply or contains a special confirmation code or password. As an email address may serve as a unique key to identify users both between sessions and across sites, it is important that this information not be provided. The difficulty here lies in realizing a solution that satisfies the following four criteria: the generated random email address must adhere to standard internet mail protocols, the system must insure delivery of confirmation mails back to the user, the system must function in a variety of different client scenarios (static IP, dynamic IP, network-address translation, etc.), and aggregation across visits must not be permitted.

The Mail-Responder module functions is installed locally on the client machine in conjunction with PAA**.** When site registration requires a valid email address for confirmation, a temporary email address is generated that routes to the Mail-Responder, a special-purpose SMTP receiver (a different mechanism is needed for machines behind NAT or running mail servers – see future work). In this scenario the user need not relinquish their actual email address, neither to the visited site nor to the PAA system itself. Perhaps most importantly there exists no local or remote mapping that can be used to link randomly generated email addresses with actual user addresses or names.

4. An Example

To clarify, we will now provide a typical example of the PAA system at work. Imagine that a user wishes to read an article from the New York Times (http://www.nytimes.com), yet does not want to provide all of the personal information required on the mandatory registration page (Figure 1). With PAA enabled, the users must simply complete form fields with short identifier strings in the format of '\[0-9]+'. As shown in figure 1, the identifiers \1, \2 and \e are typed to complete as the portion of the form that the user feels is appropriate. '\e' is a reserved identifier that triggers the substitution of a random valid email address that will route back to the Mail-Responder component. All other inputs may be completed as usual with user preferences to facilitate in-session personalization as desired. On the users first visit to the site and submission of the form, the data is sent to the AFA via HTTP POST with the following string included in the header:

```
URI=\&OQ=\&login=\%5C1\&PAAswd1=\%5C2\&PAAswd2=\%5C2\&e
mail=\%5Ce\&gender\_check=M\&birth\_year=76\&zip=11333\
&country=US\&income\_select=5\&industry\_select=3\&titl
e\_select=9\&function\_select=7\&paper\_select=2\&today
```

```
sheadlines\_check=Y\&format\_check=H\&cmpgn\_max=10\&Re
gister=Click+to+Register
```

This string is stored in PAA local database, with identifiers denoting sensitive data – the \1, \2 and \e seen earlier – are replaced with random strings of the appropriate form. It is worth noting that hand-entry of 'personalizable' fields (those that denote special interests or content areas) need only occur once per site as these too are stored in the local database. On subsequent visits to the Times registration page, the PAA Sign-In Button (see arrow below in figure 1) is added to the requested page. When clicked, the specified form fields will be auto-completed with new random strings and hand-entered fields will be re-populated with data from the previous visit. A new user is created as far as the site is concerned, while still preserving 'personalized' fields, and requiring minimal effort by the user.

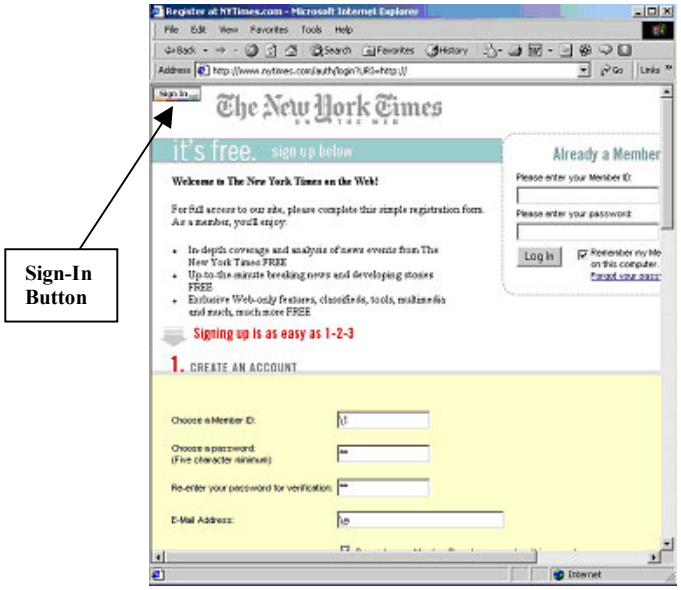

Figure 1. NYT Registration form w' PAA Sign-In Button

5. Related Work

In the table below we compare PAA to other software systems that protect privacy across four dimensions. The first two dimensions refer to types of online anonymity: *IP Anonymity*, which protects the identity of the user by disguising the IP address of her machine, and *User Anonymity,* which reflects a combination of how well the system protects a) against the collection and aggregation of information obtained via cookies, and b) against information the user yields (name, address, email, etc.) as a condition of entry, i.e., what users would fill out in registration forms. The third dimension addresses the impacts of a system on possible personalization and customization benefits. Additionally, a fourth dimension, '*Third-Party Independence'* has been added to capture the number and types of third parties that must be trusted for the proper functioning of the system. To clarify, if a user must trust a single or small group of third party-servers in a given system, for example, then its score is decreased as this server may be compromised or become unavailable without the user's knowledge. Similarly, if a central database were to exist that mapped user aliases to their actual data or names, then the score for that system would be decreased as such a database provides additional vulnerability beyond users control. If trust is distributed across a large group of users and systems, then that system might receive a medium score, as a high number of different individuals must be trusted, but each only

to small degree. Table 1 compares the capabilities of several systems for providing IP Anonymity, User Anonymity, Personalization, and Third-Party Independence.

System	IP Anonymity	User Anonymity	Personalization	Third-Party Dependency
Anonymizer	Low	Med.	n/a	Low
Onion Routing	High	n/a	n/a	Med.
Crowds	High	n/a	n/a	Med.
LPWA	Low	Med.	High	Low
PAA	Low	Med.	Med.	High

Table 1. A Comparison of Privacy-Protecting Software Tools.

The Anonymizer [23] is a commercial proxy service which provides limited *IP Anonymity*, medium *User Anonymity*, and no explicit *Personalization*. It functions as an intermediate entity, filtering HTTP headers and removing Java and JavaScript for web browsing. It rewrites all HTTP pages so that clicking on any link causes an initial request to be sent to the Anonymizer server, which in turn issues the actual request to the website. The Anonymizer currently has no features provided for anonymous registration, and hence no simple and secure means for users to preserve *User Anonymity* at web-sites offering personalized services[48]. Additionally, *Third-Party Independence* is weak in this system as trust must be invested in one or few third-party servers. Finally, the removal of Java and JavaScript renders some number of sites unusable.

Onion Routing [36,29] and Crowds [28] are two systems that provide a high degree of *IP Anonymity* for web browsing. Onion Routing transforms a message into several layers of encryptions, or "onions", with each layer determining the next forwarding node or "onion router". To enable two-way communication, onion routers maintain connection state. Crowds randomly assigns a native route for each crowd member's before the connection is routed outside the crowd[48]. While these systems have no mechanisms for *User Anonymity*, they do score better on *Third-Party Independence* than other systems. Although control does not reside primarily with the user as in PAA, its distribution over a large and dynamic group of 'third-parties' provides better protection than the centralized systems discussed earlier. We also note that PAA can be combined transparently with a system like Onion Routing to provide a high degree of *IP Anonymity* while still requiring little dependence on third parties.

Lucent's Personal Web Assistant (LPWA), provides filtering for *User Anonymity* while maintaining a high degree of *Personalization* within a single site. It also provides limited *IP Anonymity* by employing a remote HTTP proxy. *Third-Party Independence* is poor in this system as trust must be invested in a centralized third-party server. Additionally, tracing all communication to and from the central proxy server may reveal the user's IP address and identity[48]. Finally, similar to PAA in its current version, LPWA does not filter Java or JavaScript, which may leak information from the browser back to the server.

Conclusion and Future Work

Phantom Access Agent (PAA) enables anonymous registration to web sites that require it on condition of entry without revealing personal information. A central feature of PAA is that it operates on the client-side, placing genuine control in users' hands for this aspect of online interaction. Increasing the quality and extent of control over release of personal information results in meaningful and more autonomous choices over when and under what conditions personal information will be shared. In the current climate, where few websites can be trusted to hold such information in strict confidence, it is quite reasonable that users would wish to maintain

discretion over who they share such information with. In the past, users who wanted to maintain their privacy might simply have had to avoid such websites, or may have faced the onerous task of providing false information on each visit. PAA offers a convenient way to overcome this information leakage problem and thus offers greater security through privacy. Although other tools perform similar functions, PAA enhances security by running the on from users' personal computers, rather than engaging remote, third-party software and systems.

Although we have succeeded in completing a functioning prototype, there are a number of ways in which PAA could be improved, particularly if it is to serve the needs of a wide spectrum of users with little technical expertise. Such improvements might include a more intuitive installation process, fully-automated form completion, and more appropriate user feedback mechanisms. Other improvements would also be needed to extend PAA's performance in a number of important directions. These include support for the HTTPS protocol, user-managed cookie-blocking and expiration, mail support for users behind network-address-translation layers, the elimination of remote DNS servers for email routing, as well as mechanisms for IP anonymity.

References

[1] Initiative on Value Sensitive Design, http://www.ischool.washington.edu/vsd/
[2] Bugnosis: http://www.bugnosis.org
[3] Gator: http://www.gator.com
[4] Roboform: http://www.roboform.com
[5] Junkbuster: http://www.junkbusters.com/ht/en/ijb.html
[6] Cookiesweeper: http://www.cookiesweeper.com
[7] IdcidePrivacy: http://www.idcide.com
[8] CookiePal: http://www.kburra.com/cpal.html
[9] CookieCruncher: http://www.rbaworld.com/Programs/CookieCruncher/index.shtml
[10] CookieCrusher: http://www.thelimitsoft.com/cookie
[11] CookieCutter: http://www.howto-central.com/cookiecutter.htm
[12] Freedom: http://www.freedom.net/?partner
[13] GuideScope: http://www.guidescope.com/products
[14] CookieCommander: http://www.virtualstar.net/netsearch/index.html
[15] CookieEater: http://www.dittotech.com/Products/CookieEaterSurfSecret
[16] CookieCop: http://downloads-zdnet.com.com/2001-20-0.html?legacy=zddl
[17] CookieViewer: http://www.winmag.com/columns/powertools/ptcookie.htm
[18] CookieWebKit: http://www.cookiecentral.com
[19] CookieMaster: http://www.barefootinc.com
[20] Buzof: http://www.basta.com
[21] NoCookie: http://www.onepointoh.com
[22] CookieTerminator: http://www.4developers.com
[23] Anonymizer: http://www.anonymizer.com
[24] Naviscope: http://www.naviscope.com
[25] "How Web Servers' Cookies Threaten Your Privacy": http://www.junkbusters.com/cookies.html
[26] "Personal Privacy Solutions, A look at Privacy Enhancing Technologies Available to Consumers", Information Technology Industry Council: http://www.itic.org
[27] Gabber E., Gibbons, P.B., Matias Y., and Mayer A. 1997. How to Make Personalized Web Browsing Simple, Secure, and Anonymous, *Proceedings of Financial Cryptography '97*, 17-31.
[28] Reiter, M.K., and Rubin, A.D. 1998. Crowds: Anonymity for Web Transaction", *ACM Transactions on Information and System Security*, 1(1): 66-92.
[29] Michael Reed, M., Syverson, P., and Goldschlag, D. 1998. Anonymous Connections and Onion Routing", *IEEE Journal on Selected Areas in Communications*, 16(4): 482-494.
[30] Friedman, B., Howe, D.C., and Felten, E. 2002. Informed Consent In the Mozilla Browser: Implementing Value-Sensitive Design, *Proceedings of the Thirty-Fifth Hawaii's International Conference on System Sciences*.
[31] Felten, E. and Schneider, M.A. 2000. Timing Attacks on Web Privacy, *CCS*.
[32] "Preventing unauthorized access to your data": http://simplyquick.com/safe-shopping.html.
[33] Freeman, M.J. "Design and Analysis of an Anonymous Communications Channel for the Free Haven project", Thesis paper.
[34] Song R., and Korba L. 2002. Review of Network-Based Approaches for Privacy, *Annual Canadian Information Technology Security Symposium*.
[35] Song R., and Korba L. 2001. Anonymous Internet Infrastructure Based on PISA Agents, *NRC Report: ERB-1090, NRC No.44899*.
[36] Goldschlag, D., Michael Reed, M., and Syverson, P., 1999. Onion Routing for Anonymous and Private Internet Connections, *CACM*.

[37] Rennhard, M., Rafaeli S., and Mathy L. 2002. Design, implementation, and analysis of an Anonymity Network for Web Browsing. *Technical Report*.
[38] Syverson, P., Reed, M., and Goldschlag, D. 1999. Private Web Browsing., *Technical Report, Naval Research Laborator.*
[39] Felten, E.W., Balfanz, D., Dean, D., and Wallach, D.S. 2001, Web Spoofing: An Internet Con Game", *Technical Report 540-96, Princeton University.*
[40] Friedman B., Hurley D. Howe D. C, Felten E, and Nissenbaum H. 2002, User's conceptions of Web Security: a comparative Study, *CHI 2002 Extended Abstracts of the Conference on Human Factors in Computing Systems*
[41] Friedman B., and Nissenbaum, H., Hurley D., and Howe D. C., Felten E., "Users' conceptions of risks and harms on the Web: A comparative study", CHI 2002 Extended Abstracts of the Conference on Human Factors in Computing Systems. New York: Association for Computing Machinery.
[42] Web-bugs: http://www.privacyfoundation.org/resources/webbug.asp
[43] Camp, L. J. (1997, February). Web security & privacy: An American perspective. ACM.
[44] Martin, D. Internet Anonymizing Tips. ;login: Magazine (The usenix association magazine) May 1998, pp 34-39.
[45] Evan Gabber, Phillip B. Gibbons, David M. Kristol, Yossi Matias and Alain Mayer, "*Consistent, Yet Anonymous, Web Access with LPWA*", Communications of the ACM, Volume 42, Number 2, February 1999, pages 42--47.
[46] Josyula R. Rao, Pankaj Rohatgi. Can Pseudonymity Really Guarantee Privacy? To appear in the 9th USENIX Security Symposium . Proceedings of the Ninth USENIX Security Symposium.
[47] Evan Gabber, Phillip B. Gibbons, Yossi Matias and Alain Mayer, How to Make Personalized Web Browsing Simple, Secure, and Anonymous (1997)
[48] Reiter, M & Rubin, A. Crowds: Anonymity for Web Transactions (1997) ACM Transactions on Information and System Security
[49] Goldschlag, Reed, & Syverson. Onion Routing for Anonymous and Private Internet Connections (1999) Communications of the ACM (USA)
[50] Camp, L. J. (1997, February). Web security & privacy: An American perspective. ACM SIGCAS CEPC '97 (Computer Ethics: Philosophical Inquiry). Previous version presented as "Privacy on the Web", The Internet Society 1997 Symposium on Network & Distributed System Security, 10-11 February 1997, San Diego, CA.