A Standard Framework for Web Personalization

Laura Thomson
School of Computer Science and IT
RMIT University
Melbourne Australia
+613 9925 9503
laura@cs.rmit.edu.au

ABSTRACT
In this paper the requirements of a standardized cross-site personalization framework for the web are discussed, and contrasted with the requirements for a federated digital identity system. A set of design goals for personalization frameworks is developed and discussed. User identification requirements for personalization are isolated and found to be more limited than in the digital identity arena. Privacy implications for personalization frameworks are discussed. Based on all of this an architecture is proposed for detailed design and implementation.

Categories and Subject Descriptors
C.2.6 [**Internetworking**]

General Terms
Design, Human Factors, Standardization

Keywords
Web personalization, digital identity

1. INTRODUCTION
According to Mobasher [1], "Web personalization can be described as any action that makes the Web experience of a user personalized to the user's taste." Generally speaking this amounts to changing the presentation and presented content of a web site according to a user's explicit or implicit preferences. This is relatively easy to do on a single web site. Explicitly you ask the user for their preferences; implicitly you study user behavior and adapt the site accordingly. This effort is repeated on many websites.

Monitoring user preferences across multiple websites is a more interesting problem. If a user visits a web site for the first time, it will be convenient if that web site is already aware of the user's preferences and/or personal information and can adapt itself accordingly. Several approaches to this have been tried: single sign on systems (SSO) [6,7], personalizing at the client side only [5], and of course spyware and ad trackers. SSO and spyware solutions have problems with user privacy. Doing all personalization on the client side is less than efficient. Also, approaches vary and no standard has yet emerged for personalization.

In this paper the requirements of a standardized cross-site personalization framework for the web are discussed, and an architecture for this framework is proposed. It is argued that the requirements for such a framework are quite different from the requirements for a federated identity system, and the two should not be confused.

In Part 2, the existing work in the area is reviewed and the rationale for this work explored. Part 3 explores the design goals of the personalization framework. Part 4 discusses techniques for user identification. Part 5 reviews what personalization data needs to be stored and shared. Part 6 outlines the suggested architecture for the overall framework, and Part 7 concludes.

2. BACKGROUND
A number of initiatives at the W3C have arisen in this area.

OPS (Open Profiling Standard) [9] is a W3C Note submitted by Netscape, Verisign, and Firefly in 1997. It presents a standard format and protocol for interchange of explicitly entered user profile data, for example, name, address, and postcode. It is not in use. This idea of exchange of user data is a useful one, however many users would not want their identifying data shared without controls upon it. For personalization, information about surfing preferences would be more useful.

PIDL (Personalized Information Description Language) [10] is a W3C Note submitted by NEC in 1999. This is a document format that stores user preferences along with content, that is, each document contains the preferences for each user of that document. This does not scale for the Web and has obvious privacy issues. It was originally suggested for use in multicasting, a technology that never became popular. PIDL is not in use.

CC/PP (Composite Capabilities/Preference Profiles) [11] is a 1999 W3C Note that is in active use. It allows mobile client device user agents to express their capabilities and preferences to a server. Although based around technology limitations of mobile devices, this type of architecture could be used as a base for user preference sharing.

P3P (Platform for Privacy Preferences) [12] is a counterpoint to any personalization system where user data is to be shared across multiple websites. This 2002 W3C Recommendation is designed to allow users to control how much of their personal data they choose to share with web sites they visit.

None of these activities enable cross-site personalization. If we consider the work of commercial vendors, however, this is a very active space. Vendors want to know about consumer preferences and this is managed very well on a single site basis by many commercial websites -- consider the personalization and recommendations performed by amazon.com [13], for example.

Studying user preferences and behavior across multiple websites has been attempted in many ways by commercial vendors. If we leave out the popular unethical approaches (ad tracking and

spyware), we are left with single sign on (SSO) systems such as Microsoft Passport [6] and Liberty Alliance[14]. These provide a federated database of user information and preferences. Users add their data to the database and it is then shared with companies that subscribe to the SSO system.

The basic problem with the single sign on approach is the issue of trust and privacy. This issue is noted as a problem even by solution vendors (for example by Sun with regard to Liberty [15]). Why should a user entrust all their details to a third party company, and hope that they will share it only when and where the user desires? The user loses control over how much of their data is shared and with whom. Newer SSOs such as Liberty Alliance[14] and SXIP [7] have improved on this issue. Liberty Alliance uses a circle of trust (CoT) mechanism instead, which reduces the reliance on and exposure to a single third party company.

SXIP allows users to have multiple *personas* (such as home, work, and anonymous) and choose which persona to share with each subscriber site. It is also Open Source so that users can see exactly how much of their information will be shared. This approach still does not overcome the basic problem with entrusting your information to third party companies.

The other problem, which led in part to the failure of Passport as a technology, is that users have no real motivation to use SSO products. Gartner [16] suggest that "Consumers didn't see the benefit, and businesses aren't going to sink any money into it until they see how it's going to increase revenue." [17]

Our goal is to design a framework for web personalization that avoids these issues with SSO. For personalization, true authentication is not necessary. We will first look at the specific design goals for such a framework and then work through the issues of user identification and user data storage.

3. DESIGN GOALS

A good framework for web personalization should meet the following design goals. (These are similar to requirements but at a higher level.) These goals have been elucidated by analysis of existing web protocols as mentioned in the previous sections. The goals have been categorized into four types for convenience, and each goal is followed by a rationale.

3.1 Functionality

The functionality goals relate to what a personalization framework should be able to accomplish, that is defining exactly what it is we are trying to do.

Goal 1: Enable personalization of web documents according to explicit and/or implicit user preferences.

This is the basic idea from which all else follows - to personalize web documents according to user needs. Explicit user preferences are users' stated preferences while implicit preferences can be deduced from observation of previous user behavior (as done for example in [1]).

Goal 2: Allow and limit identification of users as necessary to personalize documents

Personalization should be based on an individual user's needs. In order to tailor content for an individual user, we must know something about them - how much is the question.

For example, a different level of identification is required for say, credit card use or access to restricted information, than web site browsing. A personalization framework should allow the right amount of identification for the task within user controlled guidelines. This allows conformance with existing privacy standards and guidelines (e.g. [12], [15]).

Goal 3: Work within existing protocols and in the current internet architecture, but also be able to take advantage of developments, emerging protocols and events where possible

Anything we design should work with current protocols and standards. For example, we should not require complete implementation of semantic web ideas but any implementation should be extensible enough to incorporate new standards as they emerge.

Goal 4: Where user data is shared, share it using a standard format.

To encourage as wide use of a personalization framework as possible, any data to be shared about user preferences should be in a standard format: that is, a vocabulary for personalization information. This format should be open as well as standard to encourage wider use.

3.2 Privacy

The privacy goals relate to the protection of user privacy, a crucial issue for any web based system in the current social and legislative climate. A great deal has been said about user privacy (for example [7, 12, 15, 17]) and clearly it is a key concern for web developers going forward.

Goal 5: Allow users to control their personal information: what is stored, where it is stored, what is shared with each web site.

This is the key point of difference between trying to develop a standard for personalization and trying to develop a standard for federated user identification or digital identity. Personalization is focused on trying to improve the user experience while protecting privacy.

Goal 6: Work within existing privacy frameworks and laws.

Web based systems that store user data need to work within legislative guidelines as well as meeting emerging technology standards such as P3P.

3.3 Usability

The usability goals relate to usability both from an end user perspective, and from the perspective of a developer trying to implement a personalization framework.

Goal 7: Improve the user experience

Since the goal of personalization is to improve the user experience we should attempt to meet this goal. Any framework should be easy for users to use and easy for them to control on both a large and small scale.

Goal 8: Work in as many situations as possible and degrade gracefully when it does not.

This property is important for any web-based technology as this is what users have come to expect.

Goal 9: Easy to implement partially or wholly from a developer perspective

The framework should be relatively simple and modular enough that there are low barriers to take-up for developers.

3.4 Performance

Since it is likely that a personalization framework for cross-site use will have to share data across the internet, the final goal is in the area of network performance.

Goal 10: Have as little impact on bandwidth and network performance as possible

Whatever technique is used to transmit personalization data, it should minimize the negative impact on users' surfing experiences.

4. IDENTIFICATION OF USERS

In order to personalize some degree of identification of users is needed -- just what degree of identification is an interesting question.

Cutting [3] talks about a -nymity model where the degree of identification varies along a spectrum. At one end there is anonymity where nothing is known about user identity. At the other end we have verinymity where we know a user's true, verified, identity. In the middle there are varying degrees of pseudonymity, where we can for example, reliably identify a user from visit to visit but not know who they actually are, to identifying groups of users (for example, visitors who are located in a particular country).

For systems involving payments or access to secure data, verinymity is required. However, for personalization purposes we do not require this degree of authentification. Simple adaptive systems have been built that use collective user behaviour (e.g. [1,8]) as a basis for personalization. This is a fairly broad degree of pseudonymity.

For more detailed personalization we need only observed user behavior and preferences over multiple websites. This is a degree of identification significantly less than verinymity. We do not actually need to know who the user is - that is, their name and any identifying data - but need only be able to recognize them from usage to usage.

Various techniques exist for identifying users. If it is only required to identify users of a single website we can simply get them to sign up for an account or set a cookie on users' machines. However, identifying users of a single website does not give such good information about user preference as information from multiple websites. For example, a user's taste in books at an online bookstore may be unrelated to which type of news they prefer to read.

As personalization does not require true verinymity, it was initially considered to use the user's IP address as a simple transparent way of identifying users across websites.

To test the feasibility of this approach, web logs from an internal website (which required all users to be logged in) were analyzed. Usernames and IPs were stored with each request in the web server access log.

Web log access mining for single websites commonly uses IP as the identifier for a user session in this way. However, for web log access mining a single user session or transaction is being considered. Over short period of time (minutes) a single IP is strongly correlated with a single user and this can be improved by use of user agent and timing of requests [1]. However when looking at requests over a longer timespan (one month of web logs) we found no correlation between IP and username. Hence we cannot use IP as the sole source of user identification for personalization as it does not meet Goal 2, allow identification of users.

Another possibility that may be considered for user identification is cookies. While not foolproof, cookies are widely used to track a user's interaction with a single website. Banner advertising systems such as DoubleClick [19] allow tracking of user behavior across multiple websites through a single tracking cookie associated with the domain of the banner ad. This technique would enable the functionality goals but runs into problems with privacy, specifically Goal 5, allow users control of where and what personalization information is stored. The same can be said about any spyware techniques.

The obvious choice for user identification is use of a single sign on / federated digital identity system. While it would be relatively simple to add personalization to these systems -- and user tracking is usually the unstated goal of these systems -- all of these systems run into privacy concerns.

A final possibility for user identification is to store and check authentication data on the client side. It is obvious that at the client side, we can identify the user. If authentication information is all stored and checked on the user's own machine, many privacy concerns can be neatly sidestepped. In addition, only at the client side do we have perfect information about exactly which websites have been visited and which content viewed by a user.

This gives us two possibilities for user identification to the level required for personalization - federated identity or local identity. These options will be discussed further in the next section in the context of privacy.

5. PERSONALIZATION, PERSONAL DATA, AND PRIVACY

In order to personalize web documents according to a user's preferences it must first be established what those preferences are. This can be done explicitly by asking the user, or implicitly by observing user behavior, or a combination of these.

User behavior is established by a user's history of surfing habits. This history may contain some quite sensitive information that the user may not want shared or stored by others. Consider for

example the user with a sensitive medical condition, or who is looking at advertisements for a new job, or personals ads. Even though this information may be gainfully used to personalize web documents for the user, this very personalization may make the user uncomfortable.

If a federated identity system is used for personalization, users are presented with the situation that this sensitive information is going to be shared and stored by third parties. This is quite different from trusting an identity provider with personal details such as a name, address, and credit card number.

To avoid this issue again we come back to the idea of storing this personal history data -- the raw data to be used for personalization -- at the client side.

If this data is then to be used for personalization, how can this work? All personalization can be done on the client side. This allows full access to user preference information, but limits the amount of personalization that can be done. Such a personalization system could only personalize the currently viewed document (by reordering and possibly blocking material, as suggested in PIDL [10], or at best fetch and pre-cache pages the user is likely to visit next that are linked from the currently viewed page.

If a web server is aware of user preferences, then more interesting personalizations can be performed, as the server side has full knowledge of what is available on a particular website and can tailor it more appropriately to user preference. This can be accomplished with the federated identity systems and the consequent loss of privacy.

The compromise suggested is to store user identification and behavior data at the client side and transmit only summary data about user preferences to the server side. This could be implemented as a modified HTTP request, telling a web server nothing about a user's identity, but instead suggesting the topics and keywords in which this user has proved their interest. This protects the user's privacy while enabling personalization of web documents at the server side, thus following the basic security principle of least privilege.

The security principle of least privilege (or PoLP) states that any user, application, or system should have the least privilege (access to data) needed to do the allocated task. [20] For example, a personalization system needs only summary or meta-data about what users are interested in rather than the full set of user data.

6. PROPOSED ARCHITECTURE

Given these arguments, an architecture for a cross-site web personalization framework can now be outlined. The present state of this work is to define this architecture in more detail and begin implementation.

A diagram of the proposed architecture is shown in Figure 1.

There are four main sections involved in implementing this

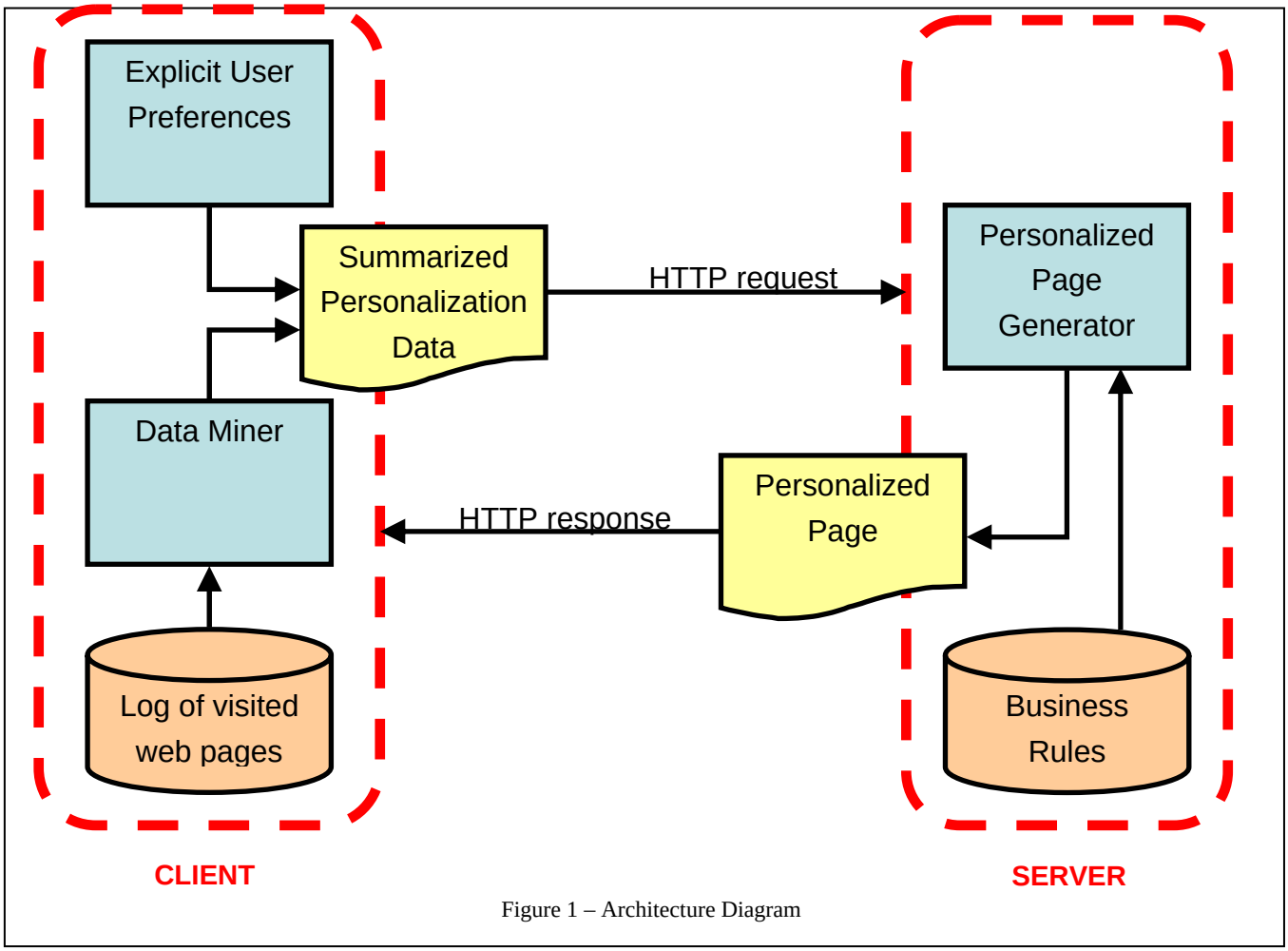

Figure 1 – Architecture Diagram

architecture. Figure 1 shows the client side identification and personal data storage (described in section 6.1), and the server side personalization (described in section 6.2). If server side personalization can not be done we can fall back to client side personalization (section 6.3). The format used for sharing user personalization information is discussed in section 6.4.

6.1 Client side identification and personal data storage

All identification of users will be done at client side via local log on. All user behavioral data will be stored at client side. This includes a full log of all pages visited, the full text of these pages, and the time spent looking at each. More sophisticated data such as gauging user interest by monitoring mouse movements (as done in some GUI HCI studies could also be included. This is the advantage of collecting the data at client side: we can get a true picture of user behaviors which is just not available at the server side.

The data collected can then be mined for summary personalization information: what this user is interested in and to what degree. This can be enhanced by collection of explicit user preference information. Explicit preference information could include, for example, explicit terms or concepts that the user is interested in or not interested in. It could also include accessibility information or information about the bandwidth capabilities of the user's connection.

The amount of data to be shared -- what and with who -- can be established by a combination of stated user preferences and privacy policies. For example, a user might choose not to share any personalization data with particular websites. Data sharing may also be controlled by user defined rules about web sites' privacy policies.

One technique that can be used to assist users in personal data management is the idea of personas. This technique is used by the SSO system SXIP[7]. Users choose whether they are in "work", "home", or "anonymous" mode and are logged in to client web sites accordingly. In the proposed framework users could define different personas to represent the type and amount of data they would like to share with different web sites. This persona information is what is to be shared with the web server. This approach satisfies the privacy-related design goals discussed previously.

Mined information should be available to users for both self censorship and persona development purposes. For example, the outcome of mining user behavior might show that a particular user is interested in investment and financial information and that they have spent a lot of time reading about a personal medical issue. The user should be able to choose not to share the fact that they are interested in this medical issue at all, or perhaps only to share that information when they are using a particular persona.

The summary personalization information generated at the client side can be transmitted to the server in several different ways. These techniques are currently being investigated, but the most promising is the idea of a "reverse cookie" where the client itself sets a cookie for a particular host as a means of transmitting data to that host. Other techniques include sending a non-standard header, making an additional POST request, or making a separate connection via another protocol such as SOAP.

6.2 Server side personalization

Based on the persona information supplied by clients as part of the HTTP request, web servers may perform personalization. The web server, if it understands the personalization information, can then choose how to act on it. If it does not understand this information then nothing will be done, that is, the system will degrade gracefully (meeting Goal 8). Exactly what and how personalization is performed is up to the individual web site. This meets design Goal 9, that is, to be easy for developers to implement.

When servers transmit a personalized page back to the client as per the request, pages should be marked as personalized, for reasons we will discuss in the next section.

6.3 Client side fallback

If this architecture came into use, it would take some time for web servers to support it. In order for users to begin to reap the benefits, there should optionally be a client side fallback personalization mechanism. While client side only mechanisms are limited in what they can do (as discussed previously) it is a compromise that works towards design goals 7 (improve the user experience) and 8 (degrade gracefully). Such systems have previously been implemented [for example, see 5] and can be used as a basis for a prototype.

When an HTTP response is received back from a web server, the client should be able to check if that page has been personalized. If this has been done, no further action is necessary, although further personalization or checking could be performed.

If the request to personalize has not been met, the client can choose to perform the personalizations (this is referred to as client-side fallback).

From the user perspective, users will be unaware whether personalizations have been performed at the client or server end.

6.4 Standard vocabulary and method for describing and sharing user preferences

This is arguably the most interesting part of this architecture and also the least developed in design at this stage. If clients are to transmit summary information to web servers, the wire format, transmission method, and personalizations required must be defined. CC/PP [11] uses a similar mechanism for sharing device capabilities so may form a basis for sharing personalization information. Whatever is implemented here must be informed by design goal 10 (have little impact on network performance). There is no point in adding to the user experience of the web only to slow it down significantly.

There is also an interesting discussion as to the level of detail recommended by personalization requests. Should such requests only list topics and keywords, for example, or suggest recommended transformations for web pages?

7. CONCLUSION

This paper has discussed the place for and requirements of a personalization framework that allows web documents to be personalized to user needs. We have recognized that the requirements for personalization are different that those for establishing digital identity and have therefore taken a different approach to the problem.

This work is at a preliminary stage. The next stage is to design in detail the components for a prototype system. This will involve the design and build of a user client (expected to be in the form of a browser plug-in), the definition in detail of the mechanism and vocabulary for sharing user preferences, and a prototype web server that can understand user preference requests. The advantage of implementing a prototype for a system that is supposed to degrade gracefully is that these modules can be developed one at a time and the system tested.

In future the detailed design and prototype system will be presented and tested for both usability and performance to see if it meets the design goals in this area. Many variants are possible and experimentation is needed to discover if this is the best way to satisfy the design goals.

8. ACKNOWLEDGMENTS

My thanks to Dr James Thom and the Web Discipline for their lively and helpful feedback on this material.

9. REFERENCES

[1] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. Communications of the ACM, 43(8):142--151, 2000.

[2] C. Payne. Everything You Need to Know About Personalization. http://www.wdvl.com/Authoring/ASP/Personalization/, 1 September 2004

[3] D. Cutting. Identity management in context-aware intelligent environments. Technical Report, Smart Internet CRC. 2004.

[4] M. Spiliopoulou, et al. A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis. INFORMS Journal on Computing, 15, 2003

[5] T. Joachims, D. Freitag, and T. Mitchell. 1997. Webwatcher: A tour guide for the World Wide Web. In Proc. IJCAI-97. http://citeseer.ist.psu.edu/joachims96webwatcher.html

[6] http://www.passport.net

[7] Dick Hardt. How SXIP Works (whitepaper). https://sxip.org/docs/specs/how-sxip-works.pdf 2004.

[8] B. Mobasher, R. Cooley, and J. Srivastava. Creating adaptive web sites through usage-based clustering of urls. In IEEE Knowledge and Data Engineering Workshop (KDEX'99), 1999. http://citeseer.ist.psu.edu/mobasher99creating.html

[9] Proposal for an Open Profiling Standard. W3C Note – 02 June 1997. http://www.w3.org/TR/NOTE-OPS-FrameWork

[10] PIDL - Personalized Information Description Language. W3C Note - 09 Feb 1999. http://www.w3.org/TR/1999/NOTE-PIDL-19990209

[11] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation 15 January 2004. http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/

[12] The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation 16 April 2002. http://www.w3.org/TR/P3P/

[13] http://www.amazon.com

[14] Project Liberty. Introduction To The Liberty Alliance Identity Architecture (whitepaper). 2003. Available from https://www.projectliberty.org/resources/whitepapers/LAP%20Identity%20Architecture%20Whitepaper%20Final.pdf

[15] Gary Ellison, Jeff Hodges, Susan Landau (2002) Security and Privacy Concerns of Internet Single Sign-On: Risks and Issues as They Pertain to Liberty AllianceVersion 1.0 Specifications. Technical Report. http://research.sun.com/liberty/SaPCISSO/sapcil.pdf

[16] http://www.gartner.com/

[17] Becker, David. Passport to nowhere? Article at news.com. http://news.com.com/Passport+to+nowhere/2100-7345_3-5177192.html?tag=nl 2004.

[18] http://www.doubleclick.com/

[19] Simson Garfinkel, Gene Spafford, Alan Schwartz (2003) Practical Unix & Internet Security, 3rd Edition. O'Reilly and Associates, Sebastopol USA.