Enhancing Web privacy and anonymity in the digital era

Stefanos Gritzalis

Information and Communication Systems Security Laboratory, Department of Information and Communications Systems Engineering, University of the Aegean, Samos, Greece

Keywords *Privacy, Worldwide web*

Abstract *This paper presents a state-of-the-art review of the Web privacy and anonymity enhancing security mechanisms, tools, applications and services, with respect to their architecture, operational principles and vulnerabilities. Furthermore, to facilitate a detailed comparative analysis, the appropriate parameters have been selected and grouped in classes of comparison criteria, in the form of an integrated comparison framework. The main concern during the design of this framework was to cover the confronted security threats, applied technological issues and users' demands satisfaction. GNUnet's Anonymity Protocol (GAP), Freedom, Hordes, Crowds, Onion Routing, Platform for Privacy Preferences (P3P), TRUSTe, Lucent Personalized Web Assistant (LPWA), and Anonymizer have been reviewed and compared. The comparative review has clearly highlighted that the pros and cons of each system do not coincide, mainly due to the fact that each one exhibits different design goals and thus adopts dissimilar techniques for protecting privacy and anonymity.*

1. Introduction

1.1 Setting the scene

The immense advances in information and communications technologies have significantly raised the acceptance rate of Internet-based applications and services. However, by allowing users to access services from virtually anywhere, the universe of ineligible people that may attempt to harm the system is dramatically expanded. Even worse, the fundamental characteristics of the Internet's communication protocol (Internet protocol; IP) itself are such that any machine on the network can monitor the IP packets transmitted and thus determine the communicating entities (sender and recipient of the packet). Even though most of the times machines connected to the Internet use dynamic IPs, implying that there is no direct association between the machine and the user, there are simple practical rules that can be applied for determining the identity of the communicating individuals (i.e. monitoring the different services that are accessed with the same IP address). It is clear that the absence of mechanisms capable to ensure the protection of privacy and anonymity can seriously affect people's communication over the Internet.

The encryption of the information exchanged between Web servers can satisfy the confidentiality requirements and deal with an eavesdropper attack. However, someone can still learn the IP address of the client and server machines, the length of the data being exchanged, as well as the time and frequency of data exchanges. Furthermore, encryption does little to protect the privacy of the client from the server, since its applicability is limited to the protection of the confidentiality of the message's content.

The author would like to thank Dr C. Lambrinoudakis and the anonymous referees for their insightful remarks and constructive comments.

Information Management & Computer Security
Vol. 12 No. 3, 2004
pp. 255-288

0968-5227
DOI 10.1108/09685220410542615

As an example let us consider a Web site that provides users with medical information and advice. Anyone can address, through the Internet, a specific request to the medical Web site and obtain the information she/he wants. Depending on the Web site, in order for the patient to access the online medical information it may be necessary to register. Usually this is an electronic process that requests from the user specific personal information. The implication is that the organization maintaining the medical Web site can easily generate "user profiles", by recording how often some specific user is visiting the site and, furthermore, the type of medical information she/he is interested in. It is therefore clear that this information can be utilized for invading the user's privacy, violating all privacy requirements as well as the 95/46 European Union directive on the protection of individuals with regard to the processing of personal and sensitive data.

Another indicative example, highlighting the importance of privacy protection, is that of modern health care environments that support electronic medical transactions (in the form of telemedicine services) between the patient and the health care organization or/and other health professionals. Normally, such medical transactions are offered through the Internet even though the exchange of personal or/and medical information is a clear prerequisite. Throughout the health care network a vast amount of sensitive medical information is being collected, stored, shared among different health care professionals and transferred to different sites worldwide. It is clear that the privacy of a patient utilizing such electronic medical services is at stake. Several unauthorized users will attempt to access the medical records of specific persons, or to analyze the information traveling over the Internet during a medical transaction of the patient, or even to steal medical information on behalf of an interested third party.

Certain malicious tactics have been employed for sacrificing privacy and anonymity over the Internet. In a traceback attack, for example, an attacker starts from a known responder and traces the path back to the initiator along the forward or the reverse path. Another class of attacks may come from a collaborator, e.g. a malicious participant in the protocol, who manages to discover the identity of some initiator. In addition, *jondos* in anonymous routing can easily launch man-in-the-middle attacks provided that the data are not encrypted on the path from the initiator to responder. Anyone who can monitor all information sent to or received by one particular protocol participant may act as an eavesdropper and record/compare all incoming or outgoing messages. In the case of an undetected eavesdropper, certain attacks such as message coding/volume attack, timing or flooding attacks, could be launched.

Message coding/volume attacks relate to cases where the contents or/and the size of a message traveling over a communication link can be traced. It is evident that an attacker, having such information available, can "link" specific communication channels / sequences / sessions with certain client-server pairs. During a timing attack, the attacker attempts to detect and analyze periodically transmitted packets, aiming to discover their source through specific time correlations. In flood attacks, one or more sources spew large numbers of packets to a target. This effectively prevents legitimate traffic from being processed, either because the target is overwhelmed with processing the flooding packets or because the legitimate traffic cannot reach the target. Furthermore, one can correlate the connection credentials of a user with the connection period to perform a homonymic attack. It is apparent that this type of attack can seriously harm individuals, compromising their privacy and anonymity. Such attacks

are normally triggered by advanced users who specifically intend to obtain information. Finally, several peer-to-peer networks have evolved through free file sharing systems. Such networks are based on applications that may contain malicious spyware or violate the privacy and anonymity of their peers (Bennett and Grothoff, 2003). Although the use of a firewall can prevent most of these attacks, there are cases where privacy violation is still a realistic scenario.

1.2 Privacy and anonymity

Privacy as a social and legal issue has, for a long time, been a concern of social scientists, philosophers, and lawyers. Privacy has been recognized as a fundamental human right in the United Nations Declaration of Human Rights, the International Convenant on Civil and Political Rights (PI/EPIC Privacy International, Electronic Privacy Information Center, 1999) and in many other national and international treaties. In democratic societies privacy must be protected. With the arrival of modern information and communication technologies systems, privacy is increasingly endangered. According to a survey presented in *Business Week* (1998), privacy and anonymity are the fundamental issues of concern for most Internet users, ranked above issues like ease-of-use, spam-mail, security and cost. According to the same survey, 78 percent of online users would use the Internet more and 61 percent of non-users would start using the Internet, if privacy policies and practices were disclosed. It is true that nowadays this specific fact is being exploited for attracting more Internet users, especially if someone considers that according to a Federated Trade Commission study, only 16 percent of corresponded sites were found to have had any privacy statement or policy in 1998, while in 2000 this percentage increased up to 65.7 percent (PricewaterhouseCoopers, 2001).

The two American lawyers S. Warren and L. Brandeis defined privacy as "the right to be alone" (Warren and Brandeis, 1890). In general, the concept of privacy can be given three aspects (Rosenberg, 1992): territorial privacy (by protecting the close physical area surrounding a person), privacy of the person (by protecting a person against undue interference), and informational privacy (by controlling whether and how personal data can be gathered, stored, processed or selectively disseminated). Certain researchers (*Business Week*, 1998; Osorio, 2001) have tried to provide an alternative definition for privacy, expressing the above-mentioned "control" of an individual in terms of: property, autonomy, and seclusion. Privacy may be understood as *property* in the sense that a person may give away part of the control over her/his personal information in exchange for some benefit. Furthermore, it may be perceived as *autonomy* in the sense that each person is free to partially or completely authorize a third party to obtain, process, distribute, share, and use her/his personal information for a specific aim. Finally, privacy may be understood as *seclusion* in the sense that everyone has the right to remain undisturbed. In this paper, we discuss informational privacy and we assume that privacy is the indefeasible right of an individual to control the ways in which personal information is obtained, processed, distributed, shared, and used by any other entity.

Anonymity and anonymous communication properties can be described along three different axes (Benassi, 1999; Osorio, 2001). The first is the type of anonymity, which is further analyzed to sender anonymity, recipient anonymity, and unlinkability of sender and recipient, where the sender is a Web user and the recipient is the Web

server. The second axis deals with the threats that each anonymity type is effectively coping with, while the third one formulates the "level" of anonymity expressing the strength of each anonymity type in a spectrum ranging from absolute privacy to provably exposed.

This paper presents some well-known Web privacy and anonymity enhancing security mechanisms, tools, protocols and services, together with the results of a comparative analysis of their characteristics. The analysis has been based on criteria that reflect users' needs and relevant privacy and anonymity requirements. The structure of the paper is as follows: a description of several Web privacy and anonymity enhancing security mechanisms, tools, protocols and services, namely Anonymizer, LPWA, TRUSTe, P3P, Onion Routing, Crowds, Hordes, Freedom, and GAP, is provided in Section 2. Section 3 describes the framework adopted for comparing different technologies taking into account architectural characteristics, operational principles and potential vulnerabilities. The comparison criteria will be grouped into three main classes: *Confronted Security Threats*, including active and passive traceback attacks, malicious collaborators, eavesdroppers, message coding/volume attacks, timing attacks, flooding attacks, connections period attacks, cookies, and personalized services provision; *Applied Technological Issues*, including reliability and trust, installation complexity, performance, overhead latencies; and *Users' Demands Satisfaction*, including connection and data anonymity and personalization, low cost, usability and friendliness, admission services provision. Finally, in Section 4 concluding remarks are briefly summarized. It is worth mentioning that there is no reference to relevant legitimate issues or protocols focused on electronic payment systems.

2. Web privacy and anonymity enhancing security mechanisms, tools, protocols, and services description

2.1 Anonymizer

Several tools have been designed for assisting Internet users to maintain their anonymity. The main objective of these tools is to ensure that requests to Web sites cannot be linked to specific IP addresses thus revealing the identity of the requestor. "Anonymizer" is a typical example of a Web anonymity tool (Cranor, 1999).

2.1.1 Introduction. The main concept introduced by the Anonymizer tool, is the existence of a third-party Web site (Anonymizer, 2003), which acts as a middle layer between the user and the site to be visited. Whenever the user wishes to visit a specific Web page, instead of establishing a direct link to the required Web server, she/he does so through the Anonymizer Web site www.anonymizer.com:8080/desired-site. Having established the connection, the Anonymizer forwards the information received from the Web site back to the user.

2.1.2 Architecture. The architecture of the Anonymizer Web tool is rather simple since it is based on the concept of a single proxy. More specifically, all Web requests issued by the users are forwarded, through the Anonymizer Proxy Server, to various HTTP servers. The requested Web pages are therefore made available to the users without leaving any trace such as IP address or any domain name system (DNS) information. Although the Anonymizer Proxy Server is considered to be a single entity, to efficiently serve a large number of Web requests it is possible to have more than one proxy or a group of proxies.

2.1.3 Operational principles. Let us consider a user who wishes to utilize the Anonymizer tool for visiting a Web page residing, for instance, at the uniform resource locator – URL www.samos.aegean.gr. At first she/he has to add the specific URL to the appropriate field in the Anonymizer's site. An alternative way is to access the required Web page through a nested URL of the form: www.anonymizer.com:8080, www.samos.aegean.gr or www.anonymizer.com:8080, www.samos.aegean.gr. In either way, the authorized Anonymizer HTTP server will forward the request of the user to the Web server and the contents of the Web page back to the user, without sacrificing in any way the anonymity of the user. This process is shown in Figure 1.

2.1.4 Vulnerabilities. It has been revealed that Anonymizer is extremely easy to use and it does not pose any requirements for modifications in the user's hardware and/or software. However, it has some inherent vulnerabilities. The first one is that the user should trust the provider of the Anonymizer service. This is because the Anonymizer server can monitor all the Web sites visited by the user, thus collecting information about her/his behavior. The second vulnerability is that although the Anonymizer is effective in hiding the identity and the IP address of the user's browser from the requested HTTP proxies, it cannot ensure that for the communication between the user machine and the Anonymizer server. Finally, the anonymity of the user can be sacrificed if "helper applications", like Real Audio, are invoked, since they bypass the proxy and establish their own direct Net connections.

2.2 LPWA

The Lucent Personalized Web Assistant (LPWA) (Bleichenbacher *et al.*, 1998; Gabber *et al.*, 1999) is a software system designed to support users in browsing the Web in a personalized, private and secure way using LPWA-generated aliases and other related LPWA features. LPWA not only generates secure, consistent and pseudonymous aliases for Web users, but also offers certain services such as e-mail support and anti-spamming filters.

2.2.1 Introduction. LPWA has been designed to address the reluctance of users to employ Web services that set as a prerequisite for their use the provision of specific personal information (Lucent Personalized Web Assistant (LPWA), 2003). The operation of the LPWA system is based on the so-called LPWA-generated aliases for Web users. Each alias consists of an alias username, an alias password and an alias e-mail address. The alias e-mail addresses allow Web sites to send messages to users, which are filtered by the LPWA system and can then be selectively forwarded to the actual user. The filtering performed by the LPWA system is based on the recipient address, which is an effective method for detecting and blocking spam.

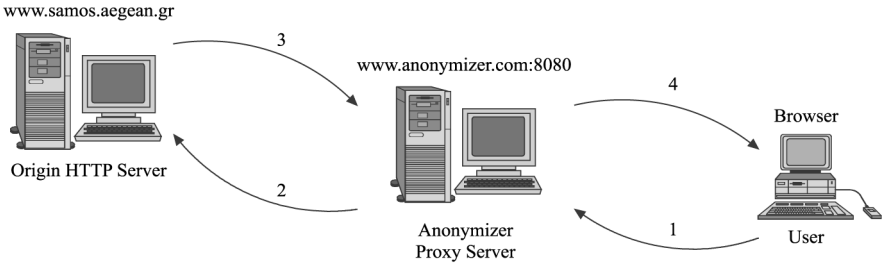

Figure 1.
Operational principles of Anonymizer

More specifically the LPWA system supports features, like automatic, secure, consistent and pseudonymous generation of aliases, e-mail service, anti-spamming support, tracing-the-traitor process, e.g. inference concerning which Web site is responsible for potentially compromising her/his e-mail address, filtering of privacy-sensitive HTTP header fields, indirection, and statelessness, since it does not keep translation tables between users and their aliases.

2.2.2 Architecture. According to Bleichenbacher *et al.* (1998) and Gabber *et al.* (1999), the LPWA system consists of three functional components: persona generator, browsing proxy and e-mail forwarder. The persona generator generates a unique, consistent and site-specific persona, as a response to a user request. The information required by the user is a user ID (e-mail address) and a secret (a password). Using these two pieces of information, as well as the address of the requested Web site, the persona generator computes, on behalf of the user, a *persona* for this Web site. The browsing proxy increases the user's privacy by in-directing the connection on the TCP level and filtering headers on the HTTP level. The e-mail forwarder forwards to the user the e-mails addressed to the persona that have been generated on her/his behalf.

The persona generator utilizes the Janus function (Gabber *et al.*, 1999), which has been designed for supporting pseudonymous client-server schemes. More specifically, the Janus function expects a UserID, a secret and a Web site domain name as input; it applies a suitable combination of cryptographic functions and produces, as output, an LPWA username and password. The current implementation of LPWA replaces certain escape sequences in the user input by the appropriate component of the alias identity.

LPWA's functional components can potentially reside at various places. The persona generator can be implemented within the user's browser or on the browsing proxy. The browsing proxy can reside on a firewall, an Internet service provider (ISP) access point, or some other Internet site. The e-mail forwarder needs to reside away from the user's machine, so as to be invisible to the user.

2.2.3 Operational principles. LPWA achieves user anonymity since aliases cannot be linked to user names. Moreover, for each Web site the user gets a different alias, thus preventing collusion of Web sites and creation of user profiles based on common keys. However, the user should be very careful about the information she/he provides to the Web sites, for instance e-mail addresses, credit card numbers, etc. that could reveal her/his identity. When the user visits a Web site that prompts for a username and a password or for an e-mail address, she/he must simply type the appropriate escape sequence (for example \u, \p, or \@), and the LPWA system will supply the Web site with the appropriate alias for the specific user. Furthermore, when the user types the escape sequence \@, the LPWA proxy creates an alias e-mail address in response. The e-mail system then stores all incoming messages and a user agent retrieves messages for all aliases that belong to a particular user. This scheme has the advantage that no privacy-compromising information has to be stored by the e-mail system. However, such a scheme is better suited for environments in which the proxy resides on a firewall or on an ISP access point.

The persona generator generates a different e-mail address for each Web site a user is visiting. This feature enables effective filtering of e-mail spamming.

The LPWA comprises the browsing proxy and the persona generator and it is stateless. The browsing proxy gets the user's identity information via an HTTP header (proxy authorization) that accompanies each HTTP request. The user's browser is

induced to start transmitting this header as part of the LPWA login process. The persona generator computes the user's aliases during transmission, utilizing the information in the header and the domain name of the destination Web site. Thus, it highlights the need to store any identity information in the proxy. Of course, the user must always log in to the LPWA with the same identity information in order to get consistent LPWA aliases.

2.2.4 Vulnerabilities. The LPWA vulnerabilities are related to the implementation schema that will be employed by the user. If the LPWA schema is the central proxy, there are serious drawbacks. Indeed, the user must trust the persons engaged with the operation of the proxy at the site www.lpwa.com. Another drawback of this specific scheme is that the connection between the browser and the LPWA proxy, which the user uses to send her/his secret, is a public connection that faces the eavesdroppers' threat. Moreover, the time for retrieving information from Web sites may be long, since every request, along with the response information, must be transferred to www.lpwa.com and backwards. Finally, LPWA does not filter JAVA and JavaScript applications, which may leak information from the browser back to the server.

2.3 TRUSTe

TRUSTe is a self-regulatory privacy initiative. Its main target is to raise the level of consumers' trust and confidence in the Internet (Benassi, 1999).

2.3.1 Introduction. TRUSTe (TRUSTe, 2003) assists the formulation of a sound privacy policy, taking into account the specific technical characteristics of the site as well as of the applications and services that they offer. The details of the privacy policy depend on the findings of the initial site review performed by TRUSTe. Subsequent reviews ensure that the operation of the Web site is in line with the privacy policy and also aim to identify modifications that may be necessary. Web sites displaying the TRUSTe trust mark, clearly indicate to their users that the information they disclose is protected in a way that has been approved and assured by a credible third party.

2.3.2 Architecture. It is evident that TRUSTe should be capable of serving trust marks and privacy statements seven days a week, 24 hours a day. In this respect, TRUSTe utilizes InterNex's dependable, fast network. In addition, redundancy is built into the system so that no single failure can affect its availability. The system components include SUN Servers, Oracle Databases, Apache Web Servers, and InterNex's Network (Figure 2). TRUSTe utilizes three servers. Two of them are located at InterNex, the one being backup of the other, while the third one is a node of the CommerceNet's network that serves as a backup routing path (redundancy) in case of Internex's unavailability.

2.3.3 Operational principles. TRUSTe conducts an initial review of each Web site's privacy policy, ensuring consistency and adherence to program principles. The aim is to assure that the privacy statements of the site reveal the types of personal information that is being gathered, who is collecting it, for what purpose and to whom it will be disclosed (Argyrakis *et al.*, 2003). After the formulation of the privacy policy, TRUSTe continues to review the site and its privacy statements periodically, to ensure that all requested criteria continue to be met. The initial and all periodic reviews are conducted at TRUSTe facility by accessing the licensee's Web site.

TRUSTe seeds Web sites with unique identifiers for tracking users' personal information. For instance, TRUSTe visits a site under an assumed identity and

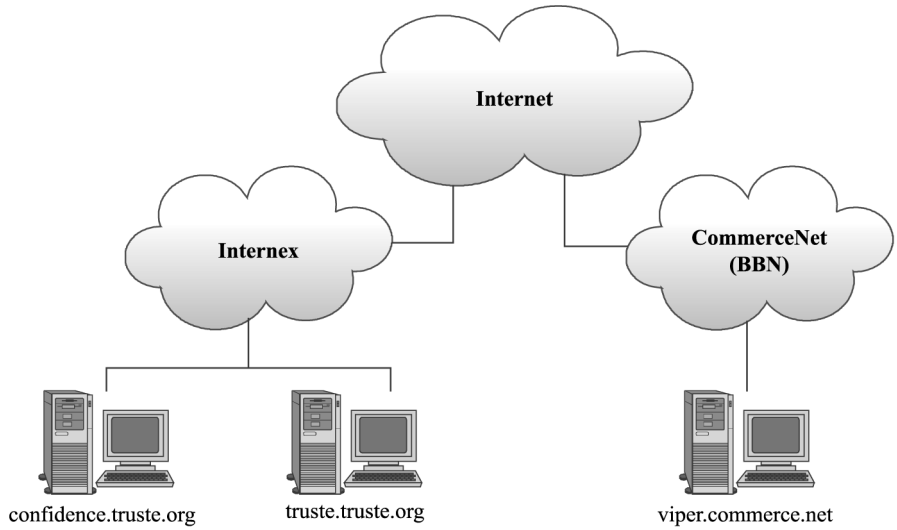

Figure 2.
A TRUSTe paradigm

provides information that has been specially compiled for the specific Web site, testing in this way the compliance with the accepted practices. The TRUSTe approval is obtained by a site if it satisfies the following conditions:

- *Notice.* The Web site must include in its home page a privacy statement that clearly describes the site's information gathering and dissemination practices. TRUSTe works with the Web site to develop comprehensive privacy statements that are easy to read and understand.
- *Choice.* The Web site must at least offer to its users the option of objecting to the distribution of their personal information to third parties.
- *Security.* The Web site must take all reasonable security countermeasures for protecting the personal data collected. Normally, the countermeasures should be the outcome of a risk analysis study that takes into account their commercial feasibility, while it is necessary to include in the privacy policy a general statement about the procedures that are in place for protecting the loss, misuse, or unauthorized alteration of information collected. If sensitive information, such as social security numbers, health information, financial account and transaction information are collected, used, disclosed or distributed, appropriate commercially reasonable practices such as encryption should be utilized.
- *Data quality and access.* The Web site must allow users, through the appropriate tools, to correct/update their personal information.
- *Verification and oversight.* TRUSTe provides assurance to users that the site complies with the declared privacy practices. If, for some reason, TRUSTe feels that a licensee is not applying the agreed privacy practices, a third party auditor may conduct a compliance review.
- *Consequence.* Depending on the severity of the privacy breach, TRUSTe may decide on revocation of the licensee's trust mark.

- *Education*. TRUSTe has ongoing education programs on privacy, targeted to consumers and businesses. Through printed and banner advertisements, consumers and Web publishers are directed to content – appropriate areas of the TRUSTe Web site.
- *TRUSTe privacy seal*. TRUSTe provides an easily recognized branded TRUSTe privacy seal to all sites that comply with TRUSTe requirements. To prevent unauthorized use of the trust mark, TRUSTe has implemented a "Click to Verify" seal. TRUSTe approved privacy statements must display the Click to Verify seal, which links to a page on TRUSTe's secured server for confirmation.

2.3.4 Vulnerabilities. The main purpose of TRUSTe is to validate the security policy elements of Web sites. Therefore, as described in Section 3, there is no protection against malicious attacks. In particular, TRUSTe service does not provide any protection from active and passive traceback attacks. As a result, the attacker could trace the initiator of a Web request. However, combining the TRUSTe's assurance with the protection offered by some other anonymizing services or tools, would result in the elimination of the problem.

2.4 P3P

The World-Wide-Web Consortium (W3C)'s Platform for Privacy Preferences Project (P3P) (World-Wide-Web Consortium W3C, 2003) provides a framework for informed online interactions. The goal of P3P (World-Wide-Web Consortium W3C, 2000) is to enable users to exercise preferences over Web site privacy practices.

2.4.1 Introduction. P3P applications allow users to be informed about Web site practices, delegate decisions to their computer agent when they wish, and tailor relationships with specific sites. Although some technologies have the ability to technically preclude practices that may be unacceptable to a user, P3P complements such technologies as well as regulatory and self-regulatory approaches to privacy. The above-mentioned technologies include anonymizers and encryption techniques, which limit the information that the recipient or eavesdropper can collect during a communication. In addition, laws and industry guidelines codify and enforce expectations regarding information practices as the default or baseline for interactions.

2.4.2 Architecture. According to World-Wide-Web Consortium W3C (2000) P3P has been designed to help users reach an agreement with Web sites, online services and applications that declare privacy practices and make specific data requests. The first step towards the agreement is that the user receives a machine-readable proposal in which the organization responsible for the service declares its identity and privacy practices. A proposal applies to a specific realm, identified by a uniform resource identifier (URI) or set of URIs. The set of possible statements is defined by the harmonize vocabulary – a core set of information practice disclosures. These disclosures have been designed for describing what a service does rather than whether it is compliant with a specific law. Moreover, some P3P implementations support a data repository where users store information they are willing to release to certain services. If they reach an agreement that allows the collection of specific data elements, such information can be transferred automatically from the repository.

2.4.3 Operational principles. P3P can be considered as a protocol for exchanging structured data. An extension mechanism of HTTP1.1 is used to transfer information

between a client and a service. At a more detailed level, P3P is a specification of syntax, structure and semantics for describing information practices and data elements, using eXtensible Mark-up Language – XML and Restricted Data Format – RDF. After reaching an agreement with a service, user agents with sufficient storage space should make note of the agreement by indexing the proposal by its propID. This allows user agents and services to refer to past agreements. Rather than sending a new proposal to the user agent on every contact, a service may send the propID of an existing agreement. In addition, P3P includes two identifiers that users can exchange with services in place of cookies. The site ID (PUID) is unique to every agreement the agent reaches with the service. If a user agrees to the use of a PUID, it will return the PUID and propID to URIs specified in the agreement's realm. The temporary or session ID (TUID) is used only for maintaining state during a single session. If a user returns to a site during another online session, a new TUID will be generated. As a result, TUID in cooperation with PUID may substitute cookies.

2.4.4 Vulnerabilities. Although users may reach an agreement with a service according to which instead of providing personal information they will simply supply a PUID or a TUID, their identity may still be traceable, for example, through their IP address. P3P acts complementary to other Web anonymity tools, protocols and services and it cannot protect the privacy and anonymity of its users on its own. However, users can eliminate this vulnerability if P3P is used in conjunction with some other anonymizing service or tool such as the Anonymizer, LPWA, Onion Routing, or Crowds.

2.5 Onion Routing

Onion Routing (Reed *et al.*, 1998) can support private communications over public networks. This is achieved through application independent, near real-time and bi-directional anonymous connections, that are resistant to both eavesdropping and traffic analysis attacks.

2.5.1 Introduction. According to Reed *et al.* (1998), packets traveling over an Onion Routing anonymous connection, instead of encapsulating source and destination addresses, only contain information about the next and previous hops. This type of connection can actually replace socket connections. Since socket connections are commonly used to support Internet-based applications, Onion Routing's anonymous connections can support a wide variety of unmodified applications through the Onion Routing proxies. Onion Routing operates by dynamically building anonymous connections within a real-time network of Chaum Mixes (Chaum, 1981), also known as *onion routers,* which are connected through permanent Transmission Control Protocol – TCP connections.

2.5.2 Architecture. Onion Routing consists of two parts: the network infrastructure that accommodates the anonymous connections, and a proxy interface that links these connections to unmodified applications

The public network contains a set of Onion Routers. An Onion Router is a store and forward device that accepts a number of fixed-length messages, performs some cryptographic transformations on them and then forwards the encrypted messages to the next destination in a random order. An Onion Router makes tracking of a particular message, either by specific bit-pattern size, or by ordering with respect to other messages, difficult. Therefore, routing a message – at the application level – through

numerous Onion Routers makes the detection of the initiator and the responder even harder. An anonymous connection is routed through a sequence of neighbouring Onion Routers.

Proxies act as interfaces between the applications and the network infrastructure. In Onion Routing, the functions of a proxy can be split into two: one part links the initiator to the anonymous connection and the other part links the anonymous connection to the responder.

2.5.3 Operational principles. After the initiator contacts her/his proxy, Onion Routing proceeds with the following steps: define the route; construct the anonymous connection; move data through the anonymous connection; destroy the anonymous connection.

More specifically, the initiator I chooses a route, through existent Onion Routers, to the responder R. For each Onion Router on the path, σ, the initiator constructs a layer of a connection setup packet consisting of the IP address of the next Onion Router, the encryption key seed information shared with the next Onion Router κ, and the successor's layer. The innermost layer of the onion contains the identity of the responder and the data to be sent. Each layer is encrypted with the public key of the corresponding router $K_{\sigma+}$. Each Onion Router pair uses a locally unique anonymous connection identifier (aci) so that subsequent communication does not require sending another onion.

$$I \rightarrow \sigma : \text{aci}, \kappa, \{\sigma', \kappa', \{\sigma'', \kappa'', \{R, \text{data}\}K_{\sigma''+}\}K_{\sigma'+}\}K_{\sigma+}$$

As the packet is forwarded through the path of Onion Routers, the layers are peeled off. When the packet reaches the last Onion Router in the path, the data are forwarded directly to the responder. All requests from the initiator are sent along the same path of Onion Routers. Replies are sent to the last Onion Router on the path, which in turn forwards the data along the reverse path of Onion Routers towards the initiator.

An Onion Router that decides to tear down a connection sends a destroy message forward and backward along the anonymous connection. An Onion Router that receives a destroy message is obliged to clean up its own table and relay the message in the same direction.

An example Onion Routing network with an anonymous connection from an initiator to a responder through Onion Routers W, X, Y, and Z is shown in Figure 3.

2.5.4 Vulnerabilities. Assuming that an attacker has collected a considerable amount of data, she/he can still analyze usage patterns and make educated guesses about the routing paths. Furthermore, if an application requires real-time communication, it may be possible to detect the socket connections that will be established between the initiator's and responder's Onion Routing proxies. One way to complicate things further and confuse attackers, is to generate dummy traffic on the network, aiming to a constant traffic level. Clearly we are now facing a trade-off between security and cost: adding dummy traffic undermines the efficiency of the Internet as a shared resource. If traffic is bursty and response time is important, smoothing out network traffic requires wasting capacity.

2.6 Crowds

Crowds (Reiter and Rubin, 1998) enable information retrieval over the Web without revealing personal information to a malicious third entity – user or computer system.

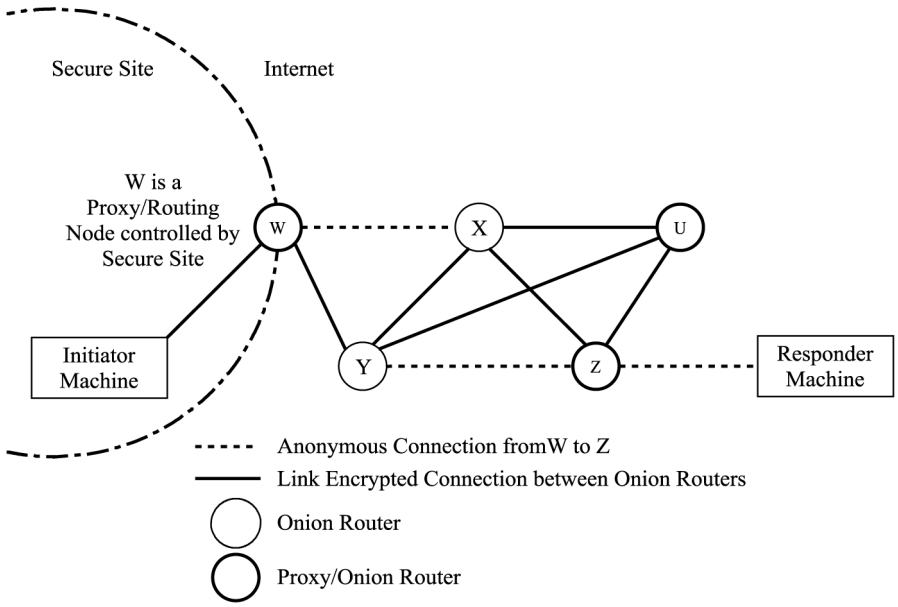

Figure 3.
An example Onion Routing network

The design aim of Crowds is to make browsing anonymous. In this way it will be possible to hide information about the user, the Internet sites that she/he visits and the content that she/he retrieves from other entities including Web servers.

2.6.1 Introduction. The principal concept of the Crowds anonymity agent is that a member of a crowd can remain anonymous and hide her/his actions within the actions of other members of this crowd. The specific protocol prevents a Web server from recording any information (i.e. domain name, user's IP address, user's computing platform, etc.) that could be potentially used for identifying the user.

2.6.2 Architecture. According to Reiter and Rubin (1998) the Crowds architecture is based on the existence of several proxies, named *jondos*, that interfere between the Crowds users and a Web server (Figure 4). A request issued by a user's browser is forwarded to the destination Web server through a number of jondos. The precise route of the request is called a *path*. It is worth stressing that an important feature of the Crowds protocol is that the request remains the same along all hops of the path and thus each jondo cannot tell whether its predecessor initiated the request or it has just forwarded it. The server's reply to the request is sent backward along the same path, with each jondo communicating with its predecessor on the path. When the originating jondo receives the reply, it delivers the information to the user's browser.

A path is modified only when jondos on the path fail or when jondos join the crowd. In the latter case, the "path memory" of all jondos is erased and all requests are rerouted from scratch. In addition, the information exchanged between jondos is encrypted, using symmetric encryption, through keys that are shared between the jondos.

2.6.3 Operational principles. Crowds concentrates Web users into a geographically diverse group, called "crowd", that performs Web transactions on behalf of its members. Each user is represented in the crowd through a process on her/his local

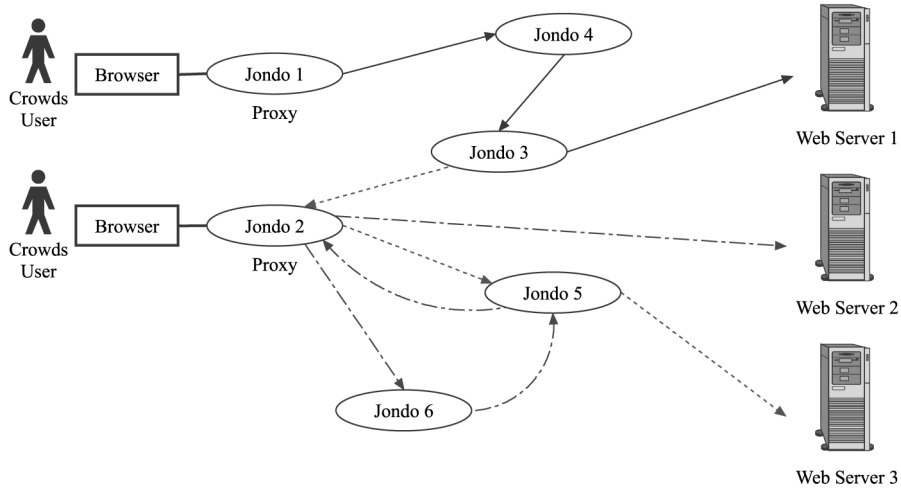

Figure 4.
Crowds architecture

machine called "jondo". More specifically, when a user starts her/his jondo on her/his local machine an automatic procedure is triggered through which the local jondo is informed about the current members of the crowd and vice versa. If this "negotiation" procedure is successful and the jondo is accepted as a member of the crowd, it can issue requests to Web servers ensuring that its identity is neither revealed to the Web servers nor to any other crowd member. In order to do this, the user configures her/his browser to employ the local jondo as a proxy for all network services, thus routing all traffic (HTTP requests) through the local jondo rather than directly to the requested Web servers. Upon receiving a request the jondo forwards it to some other, randomly selected, member of the crowd. Alternatively a jondo may decide, again in a random way, to submit the request to the appropriate Web server instead of forwarding it to a different member of the crowd.

Subsequent requests initiated by the same jondo follow the same path through the crowd, even if these requests are targeted to different Web servers. Therefore, once established, a path remains static as long as possible to prevent certain attacks on anonymity (Osorio, 2001). To achieve this, each jondo on a path records its predecessor and successor on the same path.

2.6.4 Vulnerabilities. Crowds' users should be aware of several potential risks. Theoretically any crowd member can end up submitting any request originated from within the crowd. If that happens, the Web server will record the submitting jondo's IP address as the request originator's address. The Crowds architecture does not protect the confidentiality of the requests' contents along the path that the request traverses. Nevertheless, the use of encryption, throughout the path, can rectify this problem. Like any other proxy-based anonymizing tool, Crowds can be circumvented if the user's browser downloads and executes mobile code that establishes a network connection back to the server that served it. Finally, as expected, the overall Web page retrieval times, the network traffic and the load on the machines executing jondos are increased. Simulation results, concerning the performance of the Crowds approach and thus the typical overheads imposed, can be found in Reiter and Rubin (1998).

2.7 Hordes

The Hordes protocol (Shields and Levine, 2000) engages multiple proxies for routing, in an anonymous way, a packet to a responder while it uses multicast services for routing, again in an anonymous way, the reply to the initiator.

2.7.1 Introduction. Hordes was the first protocol to take advantage of the anonymity characteristics and performance benefits inherent in the IP multicast routing. IP multicast ensures a receiver's anonymity, while it provides the shortest reverse path, thus reducing the communication latency. Multicast class-D addresses are not associated with any particular device attached to the network; instead, they are simple labels that refer to the receivers of the group as a whole, without knowledge of any particular receiver or of group membership. The number of hosts acting as receivers of a multicast routing tree, as well as their status, is dynamic and unknown to routers and hosts. These particular properties of multicast communication are clearly valuable for protecting anonymity. Hordes utilizes multicast communication for the reverse path of anonymous connections.

2.7.2 Architecture. According to Shields and Levine (2000), the Hordes architecture is based on the existence of several proxies, known as *jondos,* which are logically positioned between a Hordes user and a Web server. A request issued by a user's browser is forwarded to the destination Web server through a number of jondos. The precise route of the request is called a *path*. Although the main characteristics are the same as those of Crowds, the functionality of the two protocols is significantly different: Hordes has been designed for gathering many participants in the same reception group. Thus, using multicast for anonymous reception protects anonymity in several ways. The destination IP address encapsulated in reply packets is the multicast group IP address. It is difficult to determine the membership of the multicast group. Even if the group membership is discovered, there exists anonymity within the receiver set. Therefore, the coordination of all routers in the tree is necessary in order to determine the receiver set, which is very difficult.

2.7.3 Operational principles. Hordes protocol occurs in specific steps and its goal is to provide new Hordes members with an authenticated, fresh list of other Hordes members. First, the initiator sends the server a request to join the Hordes including the IP address of the initiator, a nonce and the initiator's public key. Then the server responds with a signed join acknowledgement that consists of a new nonce and a repetition of the initiator's nonce. The initiator replies with a signed copy of the nonces. If the nonces are correct, the server sends a multicast base address used by all Hordes members, a list of all other Hordes members and their public keys. The server ensures that the list is fresh by including the nonces, and authenticates the list by signing it. The server then informs the entire Hordes that a new member has joined the Hordes by sending a single multicast message.

2.7.4 Vulnerabilities. Hordes maintained an acceptable degree of anonymity as clearly presented in Shields and Levine (2000). However, Hordes, like any other proxy-based anonymizing service, can be violated if a browser downloads and executes mobile code that opens a network connection back to the server that served it. Moreover, passive traceback attacks are easier when targeted to centralized networks rather than to a group of widely distributed hosts.

2.8 Freedom

Freedom (Boucher *et al.*, 2000) has been designed to protect the privacy of users being involved in tasks like sending an e-mail, browsing the Web, posting a message to a news-group and participating in an Internet chat.

2.8.1 Introduction. Freedom follows a client-server architecture and it works transparently with current Internet applications. When using Freedom's premium services, Internet traffic is encrypted before leaving the user's computer and routed through private connections. Furthermore, it utilizes pseudonymous digital identities, known as *nyms*. It is used in the same way as regular online identity, without revealing the identity of the user.

2.8.2 Architecture. According to Boucher *et al.* (2000), the Freedom system includes the Freedom network, the Freedom core servers and it is supported through some software components, namely the Freedom client, the Freedom Anonymous Internet Proxies (AIPs) the network information and reporting system, the public key infrastructure – PKI servers and the mail system.

The Freedom network is an overlay network that runs on top of the Internet. It uses layers of encryption to allow a Freedom entity to use pseudonyms, hiding the user's real IP address, e-mail address and other identifying information from eavesdroppers and other active attempts to violate users' privacy. It consists of a set of Freedom server nodes, called Freedom Anonymous Internet Proxies (AIPs). The user selects the number of nodes used in a route by setting her/his level in the Freedom client. The AIPs themselves are not linked through a fixed topology. Instead, they can communicate with any other AIP on the network, as requested by a client when creating a route. The Freedom client is given a network topology that identifies a set of reliable links between nodes to simplify route selection. This topology is defined solely on the basis of AIP-AIP performance characteristics. Authenticated routes to an exit AIP are granted through route creation requests signed by a nym. Upon receipt of such a request, an AIP will retrieve the requesting nym's public signature key and verify the request. Routes that cannot be authenticated to a core AIP, are granted to any Freedom client that requests one, limited in that only Freedom core servers can be accessed.

The Freedom core servers, which are provided by Zero Knowledge Inc., provide necessary basic services, in order to keep the Freedom network running. It includes provision of public keys, nym creation and update, network information and reporting, token creation and the mail system. A core AIP located within Zero Knowledge's machine room allows unauthenticated connections to any of the core servers in the network.

The Freedom client allows networking applications running on a computer to access the Internet through the Freedom system. It is designed to work seamlessly with these applications by trapping and redirecting the data streams they generate. The Freedom client consists of several components (Figure 5): a GUI, a network access layer, a traffic filter, application filters and a set of libraries that implement the functionality required for encryption, routing, nym management, route creation, etc.

The Freedom AIPs are the core network privacy daemons that make up the Freedom network. They pass encapsulated network packets between themselves until they reach an exit node. The masquerading AIP is incorporated into the Linux kernel to access the kernel's networking features. This allows the AIP to route large amounts

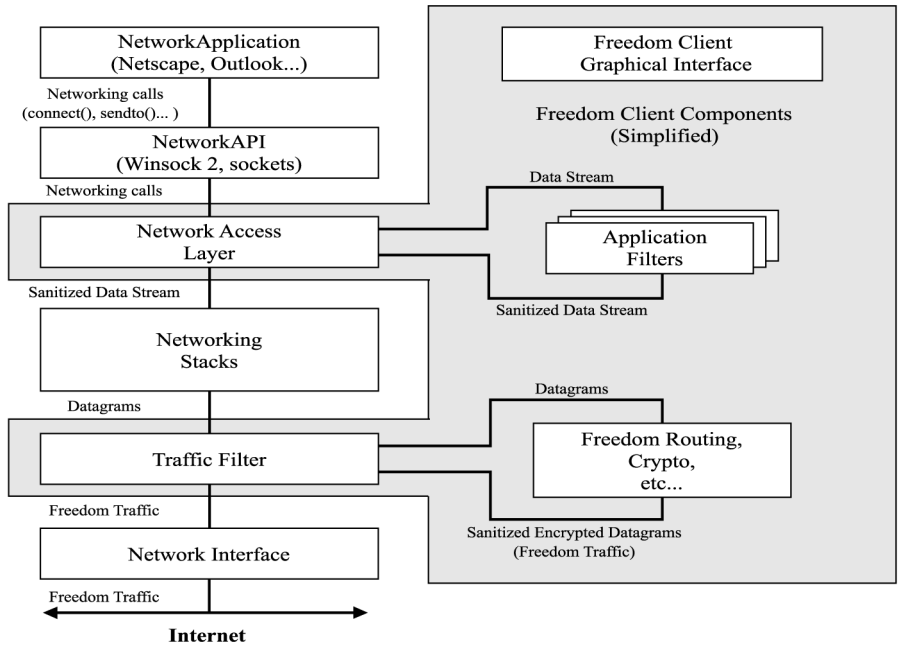

Figure 5.
Freedom's client architecture

of network traffic efficiently. A user-level daemon handles reliability-demanding operations such as key negotiations.

The network information and reporting system is made up of nodes that contain three different components: a NISS server which receives status updates from server entities, a NIDB server which stores the information, and a NIQS server that mines the database to answer network information queries for Freedom system entities.

The public key infrastructure – PKI servers consist of three server types: a key query server is used to provide public keys for ensuring confidentiality and integrity during requests to Freedom servers and clients; a nym server used to create and manage a Freedom client user's pseudonymous identities; and a token server that allows users to anonymously create nyms, without identifying their real identity.

The Freedom mail system handles all mail sent to and received by nyms. It uses the Freedom network to anonymously deliver mail messages and uses a POP box in the Freedom core to store mail that will be read by nyms. There are three entity types for the mail system: the POP entity, which is the Freedom POP server; the IMEP – Internet mail encryption proxy entity, which is used to process mail to a nym; and the NMTA – nym mail transfer agent entity, which is used to process mail from a nym.

2.8.3 Operational principles. In order to protect users' privacy, the Freedom users are encouraged to create pseudonyms, named nyms, for every activity area where they wish to preserve their privacy. The nyms that someone uses cannot be tied together. Freedom protects users' privacy by proxying the supported protocols and transmitting the proxied packets through a private network, before they are deposited on the Internet. This private network, as a system, is operated by Zero-Knowledge Systems, Inc.

Individual nodes on the network are operated either by Zero-Knowledge or by other partners, so that there is no single operator with comprehensive knowledge of the data traveling over the network. The Freedom network transports encrypted IP traffic from one node to the next.

The node's operator generates the signature and encryption keys and submits the public components to Zero-Knowledge Systems for distribution to the rest of the network. Zero-Knowledge Systems never sees the node's private keys, ensuring that only that node is capable of decrypting its layer from the traffic that passes through it.

2.8.4 Vulnerabilities. A possible way to attack a user would be to send her/him a large e-mail and then, when she/he connects via the Freedom network to the mail system to retrieve the mail, try and track back the connection to the user. It is possible that a very powerful attacker could compromise machines, or record traffic at various network access points, to identify the user.

2.9 GAP

GNUnet's Anonymity Protocol GAP (Bennett and Grothoff, 2003) is a recently presented protocol claiming to achieve anonymous data transfers. It is important to stress that GAP engages a new perspective on anonymity. Specifically, it is possible for an individual to ignore some of the traditional anonymity requirements and thus allow network nodes to balance anonymity with system efficiency.

2.9.1 Introduction. GNUnet is a peer-to-peer network and thus there is no node controlling the network. The GNUnet framework provides peer discovery, link encryption and message batching. GNUnet's primary application is anonymous file sharing, using a content encoding scheme that breaks files into 1K blocks. These blocks are transmitted using GAP. The communication between all network nodes is confidential; no host outside the network can observe the actual data traveling across the network. Even the data type is not identifiable since all packets are padded in order to have identical size.

2.9.2 Architecture. GAP aims to hide the identity of an initiator and a responder from all other entities, including GNUnet routers, active and passive adversaries and the responder or initiator, respectively. However, GAP does not strive to hide participation in the protocol, since it is only important that no adversary can correlate an action with the initiating participant.

A node, using GAP, can specify the required "anonymity level" and trade anonymity for efficiency. It is vital to stress that the decisions of individual nodes do not affect in any way the protection of anonymity and the overall efficiency of the GNUnet. This is due to the obscuring network traffic technique that is used on each node and to the fact that the protocol permits direct connections between senders and receivers. In order to determine the required anonymity level it is necessary to consider the potential threats and the capabilities of the adversary, which essentially must be guessed.

GAP has been designed for a secure peer-to-peer network, the GNUnet. Clearly it complies with principles and fulfills the requirements of this network, permitting transactions only between peers inside the borders of GNUnet. The most significant difference between GAP and prior mix-based protocols is that traditional mix protocols, such as Crowds, always perform source rewriting at each hop. GAP mixes

can specify a return-to-address other than their own, thereby allowing the network to route replies more efficiently.

GAP does not avoid a direct network connection between the initiator and the responder. In order to ensure anonymity, it is only important to decouple the relationship between the initiator and the action. Clearly, an individual cannot be identified if an adversary cannot determine the initiator of an action. This can be achieved by making the initiator look like an intermediary: a participant that merely routes data. This realization allows GAP to bypass a typical restriction of most in-direction-based anonymous routing protocols, which require that either the reply follows exactly the same path as the request or the path is statically predetermined and cannot be optimized. Hordes uses multicast communication for the reply, which however consumes large amounts of bandwidth.

2.9.3 Operational principles. Two types of messages are supported by the GAP protocol: queries and replies. A query consists of a resource identifier (RI) and a node identifier (NI) that specifies where the reply should be addressed. A reply is merely the data that were requested. Communication between nodes uses link encryption. Thus, each node can be linked to many nodes.

The RI is passed to the GAP query from the application layer. If a reply has not been received in a specific time window, the query is retransmitted. The application layer is responsible for deciding when to abandon the operation.

After a query is received by a node, it is processed according to the following steps (Bennett and Grothoff, 2003):

(1) Determine whether the specific node is available for processing the query; the decision depends on the CPU and bandwidth availability, the priority of the query, and the potential free space in the routing table. If the node is unavailable, the query is dropped.

(2) Determine whether the desired resource is locally available. If it is, enqueue the reply into the sender queue of the receiver that was specified by the query.

(3) Decide to how many nodes n the query should be sent; if $n > 0$, enqueue for sending to n other nodes.

(4) Flush individual queues, containing a mix of queries and replies, after a random amount of time.

(5) When receiving a reply, look in the routing table for the matching query or the identity of the next receiver. Enqueue the reply or pass the content to the application layer. If the content is considered valuable, copy it to the local storage, provided that there is space available.

(6) Discard rarely accessed content from local storage.

(7) Send some random content out into the network in order to generate some background noise when network is idle.

2.9.4 Vulnerabilities. GAP is a protocol that has been customized to the functionality of a peer-to-peer network. This may prove to be its greater vulnerability. Indeed, it protects the anonymity of the peers using obscuring network traffic techniques but it cannot guarantee reliable delivery of packets since the buffers (local storage) may be discarded in case of a busy node or high network traffic. Moreover, the more traffic a node creates, the more foreign traffic it must route in order to obscure its own actions.

Nevertheless, while trading anonymity for efficiency may mean being less anonymous in the short term, increased network efficiency may mean greater usability. This may lead to a significant increase of the number of users and therefore eventually to increased anonymity. On the other hand users should share data in order to be regarded as legitimate peers and be awarded various privileges, something that contradicts certain privacy requirements such as seclusion.

2.10 Conclusions

Up to this point, several different approaches for achieving Web privacy and anonymity have been presented. One of the approaches was to interpose a new entity between the sender and the receiver, in order to hide the identity of the server. Examples of such proxies are the Anonymizer and the LPWA. A different approach for anonymous Web transactions is to use enhanced proxies, which in addition to hiding the sender from the receiver they can also protect both the server and the receiver from eavesdroppers. Examples include Onion Routing and to some extent, Crowds and Hordes. Privacy can be also protected at the communication protocol level. An indicative example is the GAP protocol, that has been designed to ensure anonymous data transfers in peer-to-peer networks. A more general purpose "privacy protection suite" is Freedom, which protects, in a transparent way, Internet users and applications. Finally, there are self-regulatory privacy initiatives, like TRUSTe, aiming to validate the security policy elements of Web sites, and frameworks, like P3P, enabling users to exercise their preferences over Web site privacy practices.

3. Web privacy and anonymity enhancing security mechanisms, tools, protocols, and services comparison

The description of the most common Web privacy and anonymity enhancing security mechanisms, tools, protocols and services, provided in Section 2, revealed the following:

- there is a significant diversion of design goals;
- depending on the technology employed, different privacy and anonymity issues are addressed; and
- different techniques are employed for achieving dissimilar objectives.

This section attempts to normalize the above-mentioned differentiations, by performing a comparative analysis across three distinct "requirements classes": the *Confronted Security Threats*; the *Applied Technological Issues*; and the *Satisfaction of User Demands*. For the purpose of the comparative analysis, each class is materialized as a group of comparison criteria, all three of them forming an integrated comparison framework.

Our intention is the global privacy protection. Therefore it is necessary to deal with the entire range of potential attacks, irrespective of their classification in the above-mentioned categories. However, taking into account the incidents reported by several European Union Data Protection Authorities, the confronted security threats class includes the attacks that are most likely to happen (active and passive traceback attacks, malicious collaborators, eavesdroppers, message coding/volume attacks, timing attacks, flooding attacks, connections period attacks, cookies and personalized services provision) and they are thus considered as the most harmful ones.

3.1 Confronted security threats

This first requirement class aims to address the majority of the security threats against privacy or/and anonymity (active and passive traceback attacks, malicious collaborators, eavesdroppers, message coding/volume attacks, timing attacks, flooding attacks, connections period attacks, cookies and personalized services provision) that have been identified today. The selected comparison criteria assess the level of protection, against these threats, offered by each system.

3.1.1 Traceback attack. In a traceback attack (Shields and Levine, 2000), an attacker starts from a known responder and traces the path back to the initiator along the forward path or the reverse path. There are two types of traceback attacks: active and passive. In both cases the target remains the same. The attacker wishes to locate the initiator of a message by tracing the communication (forward or reverse) path. The differentiation is that during an active attack the attacker can take control of the network in the sense that she/he can trace back the origin of traveling packets, while in a passive attack she/he can somehow collect information regarding the routing properties of the protocol and the network nodes enabling her/him to locate the initiator of a message in a static way:

- *Anonymizer*. This system does not provide any protection against active or passive traceback attacks. The main flaw is that the user data sent to the Anonymizer proxy are not encrypted nor processed in any way.
- *Crowds*. Crowds is not vulnerable to active traceback attacks because of the use of jondos and cryptography. However, this is not the case for passive traceback attacks. If an attacker can engage many resources, she/he could exploit several jondos in order to obtain routing information. This information may reveal the initiator of a Web request.
- *Onion Routing*. This protocol achieves high level of protection against active traceback attack, due to the use of mixes – Onion Routers and different layers of symmetric cryptography. However, for the same reason described for Crowds, passive traceback attacks are not covered. Moreover, the design choice of static paths, for both Crowds and Onion Routing, increases the probability of this type of attack.
- *TRUSTe, P3P*. The main objective of these services is to negotiate and assess specific security policy elements, rather than providing protection mechanisms. It is therefore evident that they do not provide any type of protection against traceback attacks.
- *LPWA*. The use of a single proxy and the lack of data encryption mechanisms cannot guarantee any protection against traceback attacks.
- *Hordes*. Although the mechanism implemented by Crowds for sending messages is also adopted by Hordes, the connection paths are not static. Therefore, an active traceback attack cannot affect Hordes' forward connections, while passive traceback attacks have fewer chances to be successful since packets follow different paths and Hordes' jondos do not store any routing information. Furthermore, this type of attacks cannot be applied to backward connections due to the use of IP multicast. Indeed, it is difficult to trace the initiator in a multicast group even if the IP address of this group is known, as the size of the group protects any initiator's anonymity.

- *Freedom.* It provides a high level of protection against active traceback attacks since it uses asymmetric cryptography mechanisms and the connection path is dynamic. Freedom does not achieve the Hordes degree of protection since it does not support IP multicast. Nevertheless, passive traceback attacks become extremely difficult due to the fact that no routing information is stored to AIPs.
- *GAP.* The GAP protocol provides considerable protection against both active and passive traceback attacks. This is achieved through the obscuring network traffic technique. Moreover, GNUnet is a closed peer-to-peer network, restricting transmission of messages only to registered network members. As a result, problems with exit nodes, which communicate with the final recipient, do not arise.

3.1.2 Malicious collaborators. A malicious collaborator is a network node, which communicates with other nodes aiming to discover the identity of a message initiator (Argyrakis *et al.*, 2003). In the worst-case scenario all but one host is a malicious collaborator. In this case any packets initiated by the honest participant are totally identifiable. Since most of the reviewed protocols use different kinds of proxies that may act as malicious collaborators the prevention of this threat is selected as a comparison criterion:

- *Anonymizer, TRUSTe, P3P, LPWA.* The architecture of these systems is based on a single proxy. Therefore, only this specific proxy can act as a malicious collaborator, implying that if an attack of this type is detected legal action can be taken against her/him.
- *Freedom.* It uses multiple proxies and is viable to this category's characteristics concerning this type of attack.
- *Crowds.* This system offers a satisfactory level of protection against the specific threat although not as strong as Onion Routing and Freedom. The fact that the users can choose the jondos minimizes the possibility of having several malicious collaborators. Although no encryption techniques are employed, the random generation of the communication paths enhances the protection level. However, the fact that the paths are static makes this protocol vulnerable to attacks from a malicious collaborator located on the path.
- *Onion Routing.* The level of protection against this type of attack is very satisfactory. Indeed, the multiple encryption layers minimize the routing (path) information that an Onion proxy could obtain. Furthermore, since its proxy has knowledge for the previous and next server in the chain, a malicious collaborator cannot detect the initiator of the connection. However, in practice Onion Routing is not typically deployed at every host. Instead, a number of dedicated Onion Routers are available and thus the first Onion Router has full knowledge of all the initiators it is servicing. Should an Onion Router be corrupted, all initiators that utilize that router could be exposed. Fortunately, an initiator is provided with the option to choose the full communication path and thus exclude any potential malicious proxies.
- *Hordes.* This system exhibits a high level of protection for this type of attack. Although forward routing uses similar techniques (unicast) with Crowds, the use of IP multicast through the backward routing protects users from malicious

collaborators. Thus, a collaborator accepts the received packets only if she/he is the initiator and a malicious one cannot detect the source of the data and rejects them.

- *GAP.* It is assumed that a malicious collaborator can monitor, at all times, the full encrypted and decrypted traffic between all nodes. However, she/he cannot cryptanalyze the encrypted traffic between two nodes, since she/he does not have control of either node, e.g. she/he does not know any secret keys. Therefore, a malicious collaborator can only understand protocol related information. Conclusively, GAP protects anonymity fairly well.

3.1.3 Eavesdroppers. An eavesdropper is an attacker that can record and compare all incoming and outgoing messages and thus monitor all packets sent to or received by one participant in order to find the initiator or the receiver of each session. This attack constitutes one of the main privacy and anonymity threats:

- *Anonymizer, LPWA.* These systems cannot protect users' anonymity against eavesdroppers. Although the communication between the proxies and the Web servers is not vulnerable to eavesdropping, the connections between the users and the proxies can be easily compromised if an eavesdropper intercepts the communication line.
- *Crowds.* The existence of jondos and the fact that data are encrypted before transmission ensures that anonymity is sufficiently protected from eavesdroppers. In addition, each jondo on the path re-encrypts the forwarded packets using different cryptographic keys. The only vulnerability is in the case that the successor of the initiator collaborates with the eavesdropper.
- *Onion Routing.* The strong cryptography utilized by Onion Routing can effectively protect from eavesdroppers' attacks. Indeed, the anonymity can be sacrificed only if the Onion Routing's packets are traced throughout the entire sequence of Onion Routers, implying that the cooperation of every Onion Router on the path is required, since only the last Onion Router is aware of the responder's identity.
- *TRUSTe, P3P.* The main objective of these services is to negotiate and assess specific security policy elements, rather than providing protection mechanisms. Therefore, the exchanged data are not protected in any way from eavesdroppers, who can obtain valuable private information.
- *Hordes.* The protection of users' anonymity against eavesdropper attacks is satisfactory. Indeed, the packets forwarded from the initiator are encrypted and on top of that Hordes members use shared multicast groups so that each receives and discards traffic meant for other members. Therefore, even if a jondo is monitored it is not easy to identify the responder with whom it communicates.
- *Freedom.* The use of symmetric encryption prevents eavesdropping attacks. Even if an eavesdropper obtains several packets from a Freedom connection, she/he has to decrypt the code and determine the owner of each nym.
- *GAP.* The fact that the GAP protocol operates over a secure network, namely the GNUnet network, that protects anonymity through secure tunneling techniques and SSL – secure sockets layer connections, makes it also resistant to eavesdropper attacks.

3.1.4 Message coding/volume attacks. The contents or/and the size of any message traveling over a communication link can be traced. Therefore, it becomes evident that an attacker can "link" specific communication channels/sequences/sessions with certain client-server pairs. The prevention of these attacks is quite difficult. However, the fact that Crowds, Onion Routing, Hordes and Freedom employ encryption mechanisms, provide some minimum protection. A satisfactory solution to message volume attacks, as realized in GAP, is to pad all messages in order to achieve a common size for all of them. Although message-coding attacks cannot be eliminated, most of the existing systems provide some kind of protection:

- *Anonymizer, LPWA*. These systems do not offer sufficient protection against message coding attacks. The messages are not encrypted and thus their integrity can be easily sacrificed. However, the data cannot be correlated with the initiator.
- *Crowds*. The threat of message coding attacks is confronted by the use of cryptography. Despite the fact that users, members of a crowd, may freely process-exchange messages, any user outside the group is not able to resolve the meaning of a message.
- *Onion Routing*. The different layers of Onion's Routing's encryption techniques offer a sufficient support against message coding attacks. Indeed, the successive encryption layers protect, as a result of the difficulty in processing encrypted messages, from malicious acts performed by either Onion Routers or other hosts.
- *TRUSTe, P3P*. As already mentioned, the main objective of these services is to negotiate and assess specific security policy elements, rather than providing protection mechanisms. Therefore, there is no protection against message coding attacks.
- *Hordes*. This protocol has several similarities with Crowds, as far as the forward routing procedure is concerned. The use of pair-wise keys between two jondos prevents other members of the Hordes and outside attackers from deciphering users' data. Furthermore, the protocol provides the option of encrypting the data during the opposite direction as well as of activating multiple routing, thus increasing the security of data transmission.
- *Freedom*. There are several similarities with Onion Routing and Crowds concerning the protection against message coding attacks. All transmitted data are asymmetrically encrypted through the Key Query Server. The private keys are only known by their owners.
- *GAP*. The GAP protocol protects GNUnet's peers from message coding attacks. The encrypted 1K packets that encapsulate queries cannot be distinguished from packets containing other data, since all of them have the same size. In addition, GAP combines asymmetric and symmetric cryptography, realizing a strong link-encryption framework that is resistant to message coding/volume attacks.

3.1.5 Timing attacks. The threat of timing attacks concerns the possibility of analyzing periodically transmitted packets, aiming to perform time correlations for discovering their source. It is true that timing attacks can allow any Web site to determine whether one of its visitors has recently visited some other Web sites. What is interesting is the fact that an attacker can achieve this without the knowledge or consent of either the user or the specific site. For example, an insurance company, through its Web site,

could determine whether a specific user has recently visited Web sites that provide medical information. A common way to relate users' identities to exchanged packets is by analyzing the time stamps of each connection. The confrontation of this threat is very difficult since any host is vulnerable to timing attacks, except those participating in a secure group of computers. As a result, most Web privacy and anonymity enhancing security mechanisms, tools, protocols and services could not protect users' anonymity from this attack. However, Crowds, Onion Routing, Hordes and Freedom are able to prevent timing attacks against internal proxies. On the other hand GAP effectively handles this type of attack, by introducing random delays for the query and the reply at every step. Moreover, the choice of the query route is made by the node at a random fashion, making the timing results harder to reproduce.

3.1.6 Flooding attacks. In flood attacks one or more sources spew large numbers of packets to a target. This effectively prevents legitimate traffic from being processed, either because the target is overwhelmed with processing the flooding packets or because the legitimate traffic cannot reach the target. In the case of a router that supports n users, an attacker may send $n - 1$ packets and trace the original one back to its source. Although this attack could be prevented by authenticating each one of the senders, it is clear that the authentication necessity does not comply with the need for privacy and anonymity. Indeed, only routers demanding authentication via asymmetric cryptography can resist flooding attacks. However, most Web privacy and anonymity enhancing security mechanisms, tools, protocols and services rely on mutual trust among their participants. Moreover, Crowds, Onion Routing, Hordes and Freedom provide symmetric or asymmetric encryption of messages, which in a way is a form of user authentication. It is important to mention that Onion Routing's next generation will provide authentication via SSL. GAP, on the other hand, protects GNUnet's peers from flooding attacks by combining symmetric and asymmetric cryptography for establishing an integrated link-encryption framework. In addition, GAP attempts to confront this type of attack by deploying a countermeasure that emergently activates a process for dropping queries from nodes that generate an amount of traffic that exceeds a specific threshold value.

3.1.7 Connection periods attacks. Many users are allowed a limited number of connections and have a standard type of Web behavior. As a result, an attacker could analyze these repeated activities and obtain private information. Although this attack does not necessarily reveal a user's identity, it can dramatically decrease the size of anonymous users in a group. This applies to systems such as Crowds, Onion Routing, Hordes and Freedom where the use of groups is one of their main architectural features. GAP is protected against this type of attack as it obscures network traffic and each node is connected with as many peers as it is able to reach.

3.1.8 Cookies. Cookies are small data files, located at the user's computer, providing information about the user to Web servers that she/he frequently visits. Cookies can threaten users' privacy and anonymity since personal data may be processed by several Web sites and the users' Web behavior can be easily determined:

- *Anonymizer*. It does not allow the use of cookies for determining the user's Web behavior and activities. When a Web request is received the only information provided is the IP address of the Anonymizer proxy. However, filling in Web forms may reveal personal preferences of the user.

- *Crowds, Onion Routing, Hordes, and Freedom.* These systems offer protection against this potential threat, since the user has the option to disable cookies. In fact the cookies are filtered by the jondos. Freedom uses application filters to remove information stored in cookies and disables them.
- *TRUSTe.* The Web sites that have been approved by TRUSTe have the obligation to publicize, but also comply with, their privacy and anonymity policy. Thus, a user can decide if by accepting cookies puts in risk her/his privacy or not. Moreover, TRUSTe may take measures against untrustworthy partners in case the privacy policies have been compromised.
- *P3P.* It provides a mechanism that substitutes the use of cookies: P3P protocol contains two identifiers PUID and TUID that identify the user-page pair and the current HTTP session. If the user accepts these identifiers, they accompany her/his data during Web connections. As a result, cookies are disabled but the user is able to exchange data and use personalized services that require their use.
- *LPWA.* The pseudonyms provided by the system refer to the user's name, the password and the e-mail address. These pseudonyms replace relevant cookies' information; therefore, any user who visits Web sites and supplies only those pseudonyms is protected against cookies.
- *GAP.* This protocol does not use Web servers and personal information about users is not stored on cookies. Therefore, cookies are not applicable for the GAP protocol.

3.1.9 Personalized services. Several Web sites offer a range of "personalized services" for attracting new users. However, there is always the threat of revealing personal information during the routine registration procedure. The increasing use of such services has led the Web privacy and anonymity enhancing security mechanisms, tools, protocols and services to consider mechanisms for minimizing this new type of threat:

- *Anonymizer.* Although Anonymizer protects the user's identity, it does not offer any kind of protection against the personalized services threats. Indeed, the registration procedure of the user with a service cannot be realized through the Anonymizer server.
- *Crowd, Hordes, Onion Routing.* These protocols do not provide any kind of protection against this threat. Their main concern is to protect the connection between the browser of the user and the Web server. Thus, no pseudonyms can be generated, nor can the personal information of the user be concealed during the process of filling in a Web form.
- *TRUSTe.* The TRUSTe contribution is what assures the proper management of personal information from Web sites that support personalized services. However, this is done in a static manner and users have to choose either to trust these Web sites or to reject them.
- *P3P.* Although this protocol does not conceal users' personal information, it ensures the proper use of such information. Any P3P user has the right to know and negotiate the privacy policy of a Web site and then she/he could entrust her/his personal data if she/he is satisfied. In the case that the agreed rules are

not followed, users can take legal measures or complain about malicious use of her/his personal data.

- *LPWA.* This protocol provides a high level of protection against the threat of personalized services and in fact it encourages their use. The LPWA proxy supports the creation and storage of different pseudonyms for each Web site that the user is visiting. Therefore, users' personal information is known only by LPWA that forwards pseudonyms to personalized services.
- *Freedom.* The nym database holds all publicly available information that describes each nym. None of the stored information reveals the real identity of a nym owner. The database can be queried by anyone, but only the owner of a nym is allowed to modify a nym record. As a result, the multiple nyms that can be generated and stored in the database can be used for minimizing the risks from personalized services.
- *GAP.* This protocol is focused on a peer-to-peer network and as a result, personalized services have not been taken into account since they apply only to Web applications.

3.2 Applied technological issues

The second class of comparison criteria addresses various technological aspects of the Web privacy and anonymity enhancing security mechanisms, tools, protocols and services. For instance, reliability and trust, installation complexity, performance, and overhead latencies are critical comparison criteria.

3.2.1 Reliability and trust. This specific criterion refers to the level of reliability and trust exhibited by each of the entities involved in each Web privacy and anonymity enhancing technology, tool, protocol and service operation. Whatever technology is employed, the protection of the user's privacy and anonymity assumes that the user's identity is disclosed to at least one participating entity. Thus, a user should trust either a proxy or some other specific participant, in order to ensure that her/his messages are forwarded:

- *Anonymizer, TRUSTe, P3P, LPWA, and Freedom.* All these systems have a common operational characteristic, which is the existence of a single proxy entity between the users and the Web servers. This proxy server is trusted, since in the case of a deliberate information disclosure the users know who is responsible.
- *Crowds.* This protocol is less reliable than the afore-mentioned systems. Any Crowds user has to trust several participants, revealing to them her/his identity. Moreover, any member of the crowd can reveal data traveling over her/his own static path. However, every user has the flexibility to choose the jondos that she/he trusts and forward her/his message through trustworthy ones.
- *Onion Routing.* The reliability of this technology is extremely high. Although, there are several participants between a user and a Web server, the use of multi-layer cryptography prevents the disclosure of personal information to malicious Onion Routers. In addition, the initiator herself/himself is able to select the connection path, excluding Onion proxies at will. The only way that personal data can be compromised is by cooperation of all Onion Routers on a path.

- *Hordes*. Similarly to Onion Routing, the reliability of Hordes is judged to be satisfactory. The use of cryptographic key-pairs increases the protection of personal data from other Hordes members or any other entity. Although there is no mechanism for selecting the Hordes members, increasing their number raises the difficulty of compromising a user's identity.
- *GAP*. This protocol cannot guarantee reliability (in the GAP case we refer to reliable delivery of packets), since the local buffers may be discarded in case of a busy node or of high network traffic.

3.2.2 Installation complexity. This criterion refers to the complexity and side effects of the installation procedures. Although for most of the privacy and anonymity enhancing security mechanisms, tools, protocols and services, only minimal adjustments are required during installation, a complex procedure could prevent a user from selecting the specific technology/tool:

- *Anonymizer, TRUSTe, P3P*. No installation is required since they are offered as Web services.
- *Crowds*. The installation procedure of this protocol is extremely complex. The user has to download the source code and compile it in order to install a crowd client to her/his machine. In addition, any crowd member has to select her/his companions.
- *Onion Routing*. The installation procedure of the current version of Onion Routing is rather complex. This is mainly due to the selection of the communication path and the use of multi-layer encryption. However, it must be emphasized that a new version of Onion Routing is expected soon.
- *LPWA*. The installation of the service requires several steps. Mainly they refer to the collection of users' personal information, which is stored in the LPWA proxy. Moreover, the user should change her/his browser's settings to use LPWA's HTTP proxy.
- *Hordes*. Every potential Hordes member must download and compile the source code of the protocol and also select her/his companions. Therefore, the installation procedure requires a lot of effort, especially for users that are not familiar with similar techniques.
- *Freedom*. This is a commercially distributed Web application and its installation procedure, although consisting of several steps, is simplified since it is automated and well explained through the available help guides.
- *GAP*. The installation of the GAP protocol is also a complex procedure. As in previous cases, the user has to download and compile the source code of the protocol in order to become a member of the GNUnet. Moreover, a peer has to negotiate the desirable level of anonymity in relation to a strong adversary.

3.2.3 Performance. The performance of most privacy and anonymity protection security mechanisms/tools is judged by the mechanisms employed at the OSI network layer and also by the utilization of each connection link:

- *Anonymizer, LPWA*. These technologies use a single HTTP proxy to serve incoming Web requests and thus the communication channel is split into two connections causing a greater latency than a single TCP connection. However, a

user cannot easily detect the increased response time. The link utilization is satisfactory since only a small number of additional packets are transmitted.

- *Crowds, Onion Routing.* The established communication path is split into several TCP connections among different jondos. Therefore, the performance and the link utilization are not optimized. Moreover, jondos may store data, in case of network latencies, and decrease the group's performance.
- *TRUSTe, P3P.* These technologies provide the best performance and link utilization. Indeed, P3P and TRUSTe do not interfere with TCP connections and they do not add any traffic to existing links.
- *Hordes.* This protocol uses UDP packets for communicating data between jondos and therefore it does not suffer from the problems listed for Crowds and Onion Routing. In addition, the use of the UDP traffic prevents the re-transmission of packets offering a better link utilization. Moreover, UDP packets sent from responder to the initiator follow the shortest path leading to fewer transmission errors. The forward path has a higher latency and an increased chance for packet loss, as compared to the reverse path. Since Hordes uses highly asymmetric paths, if TCP were to be used it would limit the benefits of a shorter overall path. Consequently, Hordes succeeds in giving a better performance than Crowds and Onion Routing, since streaming content applications rely on UDP transmission.
- *Freedom.* The performance of the Freedom system depends on the use of asymmetric cryptography and the continuous queries for the public keys and nyms. The communication between the network of AIPs and the user is established through the exchange of TCP packets. The link utilization is satisfactory since it uses a single TCP connection and there are no other participants.
- *GAP.* The option offered by the GAP protocol to lower the anonymity's protection level for raising the network's efficiency has a positive impact on the performance of each node. Nevertheless, this does not affect the overall performance of the protocol.

3.2.4 Overhead latencies. The protection of privacy and anonymity may cause overhead latencies to Web browser activities. This is due to several complementary activities that are required for securing the data and the communication connections:

- *Anonymizer, TRUSTe, P3P, LPWA, and Freedom.* These Web technologies, which operate transparently, demand no further actions during Web browsing since the proxies of each system perform all necessary actions.
- *Crowds, Onion Routing, and Hordes.* Hordes causes the lower overhead latency despite the fact that it uses the same mechanisms as Crowds. Hordes' members do not perform any complex tasks during the backward routing procedure and there is no option to choose the communication path. Using multicast for a direct return path, Hordes significantly decreases the total expectation time from initiator to responder. Crowds is better than Onion Routing as it does not support complex cryptography mechanisms, such as multi-layer encryption. It is important to mention that, in general, the better the level of anonymity protection the greater the overhead latencies.

- *GAP.* Since GAP is a protocol focused on the secure GNUnet peer-to-peer network, the criterion of overhead latencies could not be easily applied, especially since the introduction of additional fields does not cause any latency to browsers' Web activities. Nevertheless, several queries are normally sent out to a group, potentially mixed with other messages such as content replies or peer advertisements. Grouping several messages to form a larger packet introduces delays and decreases the per-message overhead.

3.3 Satisfaction of users' demands

This last class of criteria aims to identify the degree to which user requirements are fulfilled by the existing Web privacy and anonymity enhancing security mechanisms, tools, protocols and services. In order to thoroughly examine this aspect, several criteria have been engaged, including connection anonymity, data anonymity and personalization, low cost, software usability and user-friendliness, and additional services.

3.3.1 Connection and data anonymity and personalization. The criterion of anonymity cannot be objectively applied to any anonymity system. This is due to the fact that each user has a different perception of anonymity; such as data anonymity, connection anonymity, and personalization. Data anonymity guards the user's credentials by protecting the data that she/he exchanges over the Internet. Connection anonymity also protects users' identity by masquerading as the communication path. Personalization refers to the ability of presenting different identities to every Web site that a user visits:

- *Anonymizer.* It offers a low level of connection anonymity since there is only one node – proxy between the user's browser and a Web server, while it does not provide any kind of personalization. Regarding data anonymity, the Anonymizer offers a medium level of protection, as the anonymity of each exchanged Web page is secured by hiding the user requests.
- *Crowds, Onion Routing, Hordes.* These systems provide high-level connection anonymity due to the use of jondos. Although Crowds and Hordes achieve medium level of data anonymity, Onion Routing provides a higher level because it uses stronger cryptographic mechanisms. However, none of these systems supports personalization.
- *P3P, TRUSTe.* These systems do not provide any kind of connection anonymity since there is no provision for rerouting information or using any type of proxies. In contrast, they support data anonymity and personalization, though the degree of success depends on the users' choice between static and dynamic paths.
- *LPWA.* The connection anonymity is low since there is only one HTTP proxy between the user and a Web server. On the other hand, LPWA achieves a medium level of data anonymity. Moreover it provides high personalization because it supports wide use of pseudonyms, thus allowing users to visit several Web sites each time presenting different identities.
- *Freedom.* This system complies with most of the user needs. It offers a high level of connection anonymity, due to the use of asymmetric cryptography, as well as of data anonymity and personalization since it uses pseudonyms ("nyms") in conjunction with cryptography.

- *GAP.* The protocol GAP achieves a high level of connection and data anonymity. Connection anonymity is achieved by hiding the identity of an activity initiator using a link-encryption infrastructure between the nodes. No scheme that tries to achieve anonymity on an observable, open network can hide the fact that a node is participating. On the other hand, GAP uses a digital envelope scheme for every node, e.g. asymmetric cryptography (RSA) to exchange 128-bit session keys that are used in a symmetric cryptography (Blowfish) scheme for encrypting data and connections. Finally, nodes periodically digitally sign and timestamp their Internet address, propagating this information with their public keys. Personalization is not applicable to the GAP protocol, due to its peer-to-peer nature.

3.3.2 Low cost. Web users insist on buying low cost tools, which, however, are of adequate quality. Most of the Web privacy and anonymity enhancing security mechanisms, tools, protocols and services are offered free of charge. However, users should pay a fee in order to get a full version of Anonymizer and Freedom. There is also a development cost associated with the implementation of applications based on protocols such as Crowds, P3P and Hordes.

3.3.3 Usability. It is evident that applications exhibiting user-friendly interfaces and being easy to use, attract users more. Although the perception of usability may depend on how familiar the user is with the Internet, a relative comparison of the reviewed tools, protocols and technologies follows:

- *Anonymizer.* The procedure that must be followed for submitting a Web request through Anonymizer is relatively easy. The users simply have to fill in the appropriate fields in the Anonymizer's home page or declare it as a proxy using the URL field.
- *Crowds.* Crowds is definitely harder to use. The user must choose her/his companions within the crowd and also specify the path that the data will follow. Moreover, the cryptographic mechanisms employed increase the complexity.
- *Onion Routing.* It is the most complex system of all, since several actions must be performed in order to encapsulate the original message using a multi-layer cryptography.
- *TRUSTe.* This is the most convenient Web anonymity service because the user should only read a comprehensible document. In this document the privacy policy of the Web site is described, so the user can choose if this complies with her/his privacy and anonymity requirements.
- *P3P.* The usability of this system can be characterized as medium, since the user must be familiar with the presented privacy policies in order to be able to make the appropriate decision. However, this procedure is necessary only during the protocol's initiation, unless the user decides to modify the established privacy and anonymity policies.
- *LPWA.* Although its installation is quite complex, LPWA is fairly easy to use. The user does not have to change the browser's interface because the LPWA procedures are transparent, but she/he must add several special characters that refer to her/his pseudonyms.
- *Hordes.* This system uses different types of routing during forward and backward connections. Therefore, all the downsides of Crowds' usability also

apply to the forward routing. Although, backward routing uses IP multicast and adds no further complexity, the forward routing and the cryptographic mechanisms that are used through the path specify the system's usability.

- *Freedom.* It offers a medium usability level, mostly because the user has to generate an asymmetric key pair based on her/his credentials. Another reason is that Freedom promotes the use of pseudonyms, which must be created by the user.
- *GAP.* It offers a low usability level since each peer has to choose a trade-off level between anonymity and efficiency. This trade-off is based on probabilistic guesses and may obscure the users. In addition, GNUnet's characteristic of file sharing determines that any participant should share several files or decide to be just a downloader and accept no privileges.

3.3.4 Additional services. The Internet offers a variety of services to its users such as e-mails, newsgroups, Web forums and others. These services are highly popular and thus several of the described Web privacy and anonymity enhancing security mechanisms, tools, protocols and services incorporate such services in order to be more attractive to users. Most of the reviewed tools or protocols (Anonymizer, Crowds, Onion Routing, P3P, TRUSTe, Hordes) do not directly support additional services (for instance Onion Routing supports Simple Network Management Protocol – SNMP and File Transfer Protocol – FTP protocols but it should be used in conjunction with other tools like LPWA). On the other hand, LPWA and Freedom use anti-spamming filters over the mail services to protect users from receiving undesirable mail. GAP simply offers efficient free and secure file sharing.

3.4 Comparison results

Table I summarizes the comparative analysis of the reviewed Web privacy and anonymity enhancing security mechanisms, tools, protocols and services.

4. Conclusions

In recent years, privacy is seriously endangered and is becoming more and more an international problem in networked society. It cannot be sufficiently protected by legislation means only. It must also be enforced by technologies and it should form a design criterion for modern information and communication systems. This paper has presented some well-known Web privacy and anonymity enhancing security mechanisms, tools, protocols and services, focusing on their architectural characteristics, operational principles and identified vulnerabilities. Furthermore, the results of a comparative analysis of these systems, across three distinct requirements classes, namely the confronted security threats, the applied technological issues and the satisfaction of user demands, have been presented. For the purpose of comparative analysis, each class consists of a group of comparison criteria, all three of them forming an integrated comparison framework.

The pros and cons of each system, in terms of the level of protection that they offer for privacy and anonymity, do not coincide. This was expected since each one focuses on different design goals. For instance, systems based on single HTTP proxies, such as Anonymizer and LPWA, can be easily employed for ensuring anonymous browsing, while systems like Onion Routing, Crowds, and Hordes are scoring better in protecting

Table I.
Comparative analysis of Web privacy and anonymity enhancing security mechanisms, tools, protocols and services

Criteria			Confronted security threats							Applied technological issues				Satisfied user demands				
Technologies	TBA (A)	TBA (P)	MC	E	MC/VA	TA	FA	CPA	C	PS	R&T	IC	P	OL	CA	DA	P	LC	U	S
Anonymizer	–	–	×	–	–	–	–	–	–	–	H	L	H	L	L	M	×	L	H	L
LPWA	–	–	×	–	–	–	–	–	+	+	H	M	H	L	L	M	H	H	M	H
TRUSTe	×	×	×	–	–	–	–	–	+	+	H	L	H	L	×	×	×	H	H	L
P3P	×	×	×	–	–	–	–	–	–	+	H	L	H	L	×	×	×	M	M	L
Onion Routing	+	–	+	+	+	–	–	–	–	–	L	H	L	H	H	H	×	H	L	M
Crowds	+	–	+	+	+	–	–	–	–	–	L	H	L	H	H	M	×	M	L	L
Hordes	+	–	+	+	+	–	–	–	–	–	M	H	M	H	H	M	×	M	L	L
Freedom	–	+	×	+	+	–	–	–	+	+	H	M	M	L	H	H	H	L	M	H
GAP	+	+	+	+	+	+	+	+	×	×	L	H	H	L	H	H	×	H	L	M

Note: H: high, M: medium, L: low, ×: not applied,+: positive, and −: negative

Key: TBA (A) = trace back attack (active); TBA (P) = trace back attack (passive); MC = malicious collaborators; E = eavesdroppers; MC/VA = message cod./vol. attacks; TA = timing attacks; FA = flooding attacks; CPA = connection period attacks; C = cookies; PS = personalized service; R&T = reliability and trust; IC = installation complexity; P = performance; OL = overhead latencies; CA = connection anonymity; DA = data anonymity; P = personalization; LC = low cost; U = usability; S = services

information and in preventing attacks. Onion Routing can form the basis upon which other protocols could be employed in order to improve the level of global user protection. On the other hand, services like TRUSTe and P3P negotiate, assure and periodically re-confirm the completeness and correctness of Web sites' privacy policies. Freedom is a system that fulfills most of the user requirements, exhibiting a satisfactory level of protection against security threats; however its high cost may downsize all these advantages. Finally, GAP can be used in secure peer-to-peer networks, such as GNUnet, to deploy secure and anonymous transactions between peers. This is extremely useful for Internet users since many of them use peer-to-peer applications that focus on file sharing (GNUtella, KaZaa, iMesh, etc.) and frequently violate the privacy and anonymity of their peers.

Technology advances and increasing users' needs, especially in commercial Internet activities, should lead to further research. The deployment of protocol, services and tool versions that support technologies, such as SSL/TLS or IPv6 characteristics, may cause the revision of this approach. Moreover, if the aim is the development of an integrated holistic security solution, Web privacy and anonymity should cover the use of digital cash and credit cards in business-to-business and business-to-consumer transactions. Additional analysis would be especially interesting in identifying features that have not been considered in the development of privacy enhancing services for e-commerce, or the major differences in the technological design and business models. Current trends towards the design and implementation of intelligent agent systems, provides motivation for analysis of integrated privacy agents systems. Finally, as invasion of privacy is not only a technical problem, but has also social, legal and psychological dimensions, a holistic approach to a privacy-friendly use and design of security mechanisms is necessary.

References

Anonymizer (2003), available at: www.anonymizer.com

Argyrakis, J., Gritzalis, S. and Kioulafas, C. (2003), "Privacy-enhancing technologies: a review", in Menzel, T. and Quirchmayr, G. (Eds), *Proceedings of the EGOV03 2nd International Conference on Electronic Government*, LNCS 2739, Springer Verlag, Berlin, pp. 282-7.

Benassi, P. (1999), "TRUSTe: an online privacy seal program", *Communications of the ACM*, Vol. 42 No. 2, pp. 56-9.

Bennett, K. and Grothoff, C. (2003), "GAP – practical anonymous networking", *Proceedings of the Workshop on PET2003 Privacy Enhancing Technologies*, also available at: http://citeseer.nj.nec.com/bennett02gap.html

Bleichenbacher, D., Gabber, E., Gibbons, P.B., Matias, Y. and Mayer, A. (1998), "On secure and pseudonymous client relationships with multiple servers", *Proceedings of the 3rd USENIX Electronic Commerce Workshop*, pp. 99-108.

Boucher, P., Shostack, A. and Goldberg, I. (2000), *Freedom System 2.0 Architecture*, Zero-Knowledge Systems, Inc.

Business Week (1998), *A Little Net Privacy, Please*, 16 March, available at: www.businessweek.com/

Chaum, D. (1981), "Untraceable electronic mail, return addresses and digital pseudonyms", *Communications of the ACM*, Vol. 24 No. 2, pp. 84-8.

Cranor, L. (1999), "Internet privacy", *Communications of the ACM*, Vol. 42 No. 2, pp. 29-31.

Gabber, E., Gibbons, P.B., Kristol, D., Matias, Y. and Mayer, A. (1999), "Consistent, yet anonymous Web access with LPWA", *Communications of the ACM*, Vol. 42 No. 2, pp. 42-7.

Lucent Personalized Web Assistant (LPWA) (2003), available at: www.bell-labs.com/projects/lpwa

Osorio, C. (2001), "A new framework for the analysis of solutions for privacy-enhanced Internet commerce", *eJETA: The eJournal for Electronic Commerce Tools and Applications*, Vol. 1 No. 1, pp. 1-8.

Privacy International, Electronic Privacy Information Center (PI/EPIC) (1999), *Privacy and Human Rights – An International Survey of Privacy Laws and Developments*, available at: www.privacy.org/pi/survey

PricewaterhouseCoopers (2001), "Privacy: a weak link in the cyber-chain", in E-Business Leaders Series, PricewaterhouseCoopers, New York, NY.

Reed, M., Syverson, P. and Goldschlag, D. (1998), "Anonymous connections and Onion Routing", *IEEE Journal on Selected Areas in Communications*, Vol. 16 No. 4, pp. 482-94.

Reiter, M. and Rubin, A. (1998), "Crowds: anonymity for Web transactions", *ACM Transactions on Information and System Security*, Vol. 1 No. 1, pp. 66-92.

Rosenberg, R. (1992), *The Social Impact of Computers*, Academic Press, New York, NY..

Shields, C. and Levine, B.N. (2000), "A protocol for anonymous communication over the Internet", in Samarati, P. and Jajodia, S. (Eds), *Proceedings of the 7th ACM Conference on Computer and Communications Security*, ACM Press, New York, NY, pp. 33-42.

TRUSTe (2003), available at: www.truste.org

Warren, S. and Brandeis, L. (1890), "The rights to privacy", *Harvard Law Review*, Vol. 5, pp. 193-220.

World-Wide-Web Consortium (W3C) (2000), "The platform for privacy preferences 1.0 specification", *W3C Candidate Recommendation 15*, W3C, Cambridge, MA.

World-Wide-Web Consortium (W3C) (2003), *Platform for Privacy Preferences Project – P3P*, W3C, Cambridge, MA, available at: www.w3.org/P3P