A survey on solutions and main free tools for privacy enhancing Web communications

A. Ruiz-Martínez[a,*],

[a]*Department of Information and Communications Engineering, Faculty of Computer Science, Campus of Espinardo, 30100, Murcia, SPAIN*

Abstract

Concern for privacy when users are surfing on the Web has increased recently. Nowadays, many users are aware that when they are accessing Web sites, these Web sites can track them and create profiles on the elements they access, the advertisements they see, the different links they visit, from which Web sites they come from and to which sites they exit, and so on. In order to maintain user privacy, several techniques, methods and solutions have appeared. In this paper we present an analysis of both these solutions and the main tools that are freely distributed or can be used freely and that implement some of these techniques and methods to preserve privacy when users and surfing on the Internet. This work, unlike previous reviews, shows in a comprehensive way, all the different risks when a user navigates on the Web, the different solutions proposed that finally have being implemented and being used to achieve Web privacy goal. Thus, users can decide which tools to use when they want navigate privately and what kind of risks they are assuming.

Keywords: anonymous communications, Web privacy, privacy, anonymity, proxies, cookies, private browsing mode

1. Introduction

In the last ten years we have assisted to an increasing interest in the research within the privacy technology field. Indeed, during these years we have observed the development on privacy solutions for different purposes: anonymous communications, identity management, languages for expressing and negotiating privacy policies, privacy preserving data publishing and mining, e-voting, location-based services, etc (Carroll and Grosu, 2009; Danezis and Gürses, 2010; Karopoulos et al., 2010). This interest within research community is also shared by end users (Gross and Rosson, 2007).

Within anonymous communications end users are interested in preserving their privacy when they surf on the Web since, currently, the access to the Web is the main use of the Internet. In fact, more and more they access more resources and, at the same time, Web sites want to know information on them since Web can be an important source of profit.

When users are surfing on the Web they are interested in protecting their *Personally Identifiable Information* (PII) from being observable. Thus, for some particular accesses, they want to be anonymous and avoid being tracked. They also want that their browsing behaviour and the sites they interact with cannot be known by observers (Chen and Fu, 2008), that is, they want to prevent the creation of profiles on them.

As users are more and more concerned on their privacy (Gross and Rosson, 2007) and want to navigate privately for many different intentions: some for legitimate purposes and some other for criminal, disruptive, or socially unacceptable purposes.

As for legitimate purposes we can mention: privacy and freedom of speech (through Webs, blogs or online social networks), anti-censhorship, anonymous tips for law enforcement, surveys (evaluation and feedback), gift shopping, obtain commercial information (query prices), protection of children privacy, query in search engines, access to pornography and the prevention organization's Web filters from monitoring or limitation of traffic bandwith (e.g., for P2P traffic that is limited for ISPs) (CISCO Systems, 2009; Aggarwal et al., 2010; Chaabane et al., 2010; Li et al., 2011).

On the other hand, as for criminal, disruptive, or socially unacceptable purposes we point out: spam e-mail, piracy, hacking, information and identity theft, cyber-

*Corresponding author. Phone: +34 868887863. Fax: +34 868884151

Email address: arm@um.es (A. Ruiz-Martínez)

stalking, exposition of an organization to malicious activities, illegal software download, child pornography, abuse of organization resources, and even for terrorism (CISCO Systems, 2009; Aggarwal et al., 2010; Chaabane et al., 2010; Li et al., 2011).

Recently, as a sample of the interest in privacy when we surf on the Web, Web browsers such as Mozilla Firefox, Microsoft Internet Explorer, Google Chrome and Safari (from Apple) have included a private browsing mode to their user interface that allows users to navigate privately. In these browsers, these modes are known as Private Browsing, InPrivate, Incognito and private browsing, respectively. However, these modes do not offer a complete solution to navigate guaranteeing privacy.

The main goals of these private browsing modes are two (Aggarwal et al., 2010). First, no to leave trace on user computer on the Web sites visited. Second, user's activity cannot be linked between the Web sites they visit and that the activities carried out in the private mode are not know in the public mode. Thus, these modes only offer a partial solution since users might be tracked, e.g., from their Internet Protocol (IP) address.

In this paper our aim is to show the process that users follow when they navigate on the Web and, from this starting point, to provide a comprehensive explanation of how the users can be tracked from the different PII can be obtained in this process as well as the different solutions and tools existing to cope with these problems.

Hence, this paper explains that this PII information can be obtained from three different conceptual layers: TCP/IP level, HTTP level and application level. Once we have presented the problems associated to each layer, we describe the different mechanisms and techniques that have been proposed up to date in order to avoid that PII can be obtained.

Although there are an important number of solutions to cover privacy in (Web) communications (Linn, 2005; Danezis and Diaz, 2008; Edman and Yener, 2009; Behl and Lilien, 2009; Danezis and Gürses, 2010; Ren and Wu, 2010), it is important to point out that we will only center in those solutions and mechanisms that have been implemented and are currently being used.

Finally, we analyse the main free tools that we have available to preserve privacy when we surf on the Web and we indicate the mechanisms they implement and the level of privacy protection they offer.

The rest of this paper is organised as follows. Section 2 describes the process that is followed when we access a Web site and the different privacy-related risks we are exposed. Section 3 introduces the solutions to overcome the risks mentioned and the main tools we can use. Once the solutions and tools have been presented, in Section 4 we compare these tools and discuss on the protection offered. Section 5 compares our work with previous works related to the analysis of solutions and tools for enhacing privacy in Web communications. Finally, Section 6 presents the main conclusions of our work and introduces future work.

2. Web navigation and privacy concerns

In this section we describe the process that is followed when a user requests a Web page. Then, we explain the information that can be obtained from a user during this process. Thus, we can understand the different privacy-related risks when we surf on the Internet.

2.1. Web page request flow

Let us suppose a user is interested in accessing the Web site of a company X. The process followed is shown in Figure 1. For this purpose, the user launches her preferred Web browser and enters the URL of the Web site (e.g. *http://www.companyx.com*). The browser sends a request to the Web site by sending a HTTP GET request (step 1). As a response to this request, the Web server sends a HTTP 200 OK response that contains the HTML page requested (step 2).

The browser processes the HTML page received and obtains the different Web objects (images, scripts, Flash objects, Activex, Java applets, Silverlight objects, stylesheets, etc) included in the HTML page downloaded (steps 2.1 to 2.4).

Some Web objects are located in the same server we have obtained the Web page. Then, the browser requests these objects to the same server by means of HTTP request as previously explained (steps 2.1 and 2.2). These steps are repeated for each object requested to this server. In this case, as a response the Web browser instead of receiving a HTML page, it receives a Web object.

Additionally, the HTML page requested by the user could contain some elements that the Web site have included and that are located in other Web sites. In this case, the browser, for each element, requests to the corresponding Web server the Web object needed. In general, these objects (usually images, pop-ups and flash objects) that a Web site includes from other Web sites (third party Web sites) are advertisements or Web bugs (tiny images of 1x1 pixels) (Rezgui et al., 2003).

There are several entities that can play the role of third party Web sites such as advertisment servers, market researchers, affiliate marketers, retargeters, third-party data collectors, etc (Gulyas et al., 2008; Toubiana

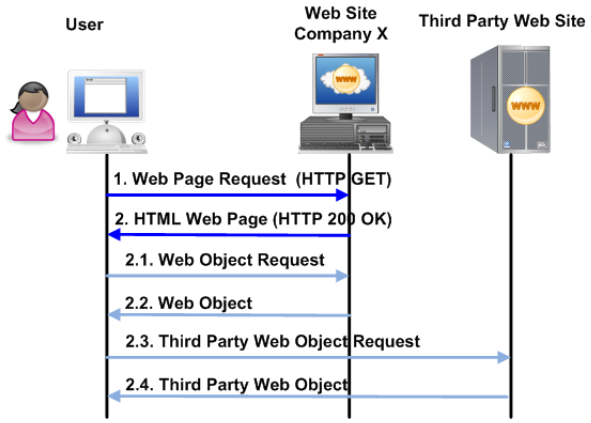

Figure 1: Web page request flow

et al., 2010; Lambrecht and Tucker, 2011). Thus, steps 2.3 and 2.4 are executed as many times as objects from these parties are placed in the HTML page.

When all the objects have been downloaded the Web browser activity finishes until the user clicks on a new link.

2.2. *Privacy concerns*

When we surf on the Web, Web sites collect information about us (usernames, email address, location information, interests, access patterns, navigating behaviour, etc) (Linn, 2005; Harding et al., 2007; Jang et al., 2010) that allows them to create profiles. Some of these important Web sites that can collect information on users are Web search engines from the search queries the users send.

The use of these profiles is twofold. On the one hand, they are used to improve the Web site and create customized services in function of user preferences, behaviour, and so on that produce a better user experience (Harding et al., 2007; CISCO Systems, 2009; Lambrecht and Tucker, 2011). On the other hand, profiles are used for making money thanks to marketing: this can attract to more advertisers (their campaigns can be more effective) and they can share this PII with other entities such as partners and affiliates (CISCO Systems, 2009; Yan et al., 2009). Namely, this information is used for targeted advertising and dynamic pricing (Gulyas et al., 2008). This latter purpose represents a user privacy threat when user is not consenting this data collection (Rezgui et al., 2003; Harding et al., 2007). Information collection is a dimension of privacy that aims that data are collected only with knowledge and explicit consent (Rezgui et al., 2003).

Next, in this section we analyse the different information that can be used to track users and create profiles about them when they are surfing on the Internet.

Conceptually, a Web user can be tracked using information of three different layers: TCP/IP layer, HTTP layer and application layer.

2.2.1. *TCP/IP layer*

HTTP requests are sent through connections that use TCP/IP (Transmission Control Protocol/Internet Protocol) protocol (Fielding et al., 1999). In this level, basically, the information that can be gathered to track the user is the IP address and the port the user is making the request. If in the user's organization Network Address Translation (NAT) is not used, the IP address identifies the particular computer (or even user) is accessing to the Web and we can link all transactions performed by this user. If NAT is used, only the IP address does not identify the user. As mentioned in (Casado and Freedman, 2007) the use of NAT is reduced and can be detected. Furthermore, complementary mechanisms that we expose in the following layers could be used.

IP address also provides domain name, geo-location information, identifying ISP, city, country, region, country and continent where the request is being made. There are many Web sites where we can access with our browser and they provide this kind of information, e.g., *showip* (Showip, 2011b). We can also find sites where, from our domain name, they can find information on our organisation (even the name of the administrator) based on the use of *Whois*, e.g., Smart Whois (AllNetTools, 2011) or Ros instrument Whois (Showip, 2011a).

In this level, the round-trip time of user's connection could also identify a user from others (Back et al., 2001; Saint-Jean et al., 2007; Hopper et al., 2007; Schlegel and Wong, 2009; Hopper et al., 2010).

The trace at TCP level can reveal, apart from the port used, information such as computers involved in the communication, uptime, operating system, NAT detection, and some other properties of the connection that are detailed in (Zalewski, 2005, 2006).

2.2.2. *HTTP layer*

HTTP is the protocol that allows us to access Web resources. It is a stateless protocol that works using the pattern request-response. A HTTP request contains the URL of the Web page to be accessed. In the HTTP response, the HTML page is received.

The HTML page received could contain links to additional Web resources needed to show correctly the Web page. This involves that new HTTP requests are performed (see Figure 1, steps from 2.1 to 2.4). These Web

resources can be images, Flash objects, CSS (Cascading Style Sheets), Javascript, VBScript, etc. Usually, these kinds of resources are recovered once the Web page is received.

In this level there are two elements that can be used to collect PII and track the user: HTTP headers and HTTP cookies (also known as Web cookies. For the shake of simplicity, hereinafter we will simply reference them as cookies). From connection established for the request we can obtain the IP and port the user is connecting to the server.

An HTTP resquest/response contains a set of headers that can compromise user's privacy. These headers could reveal the following information: user's Web browser (*User-agent* header), language, encoding and charset preferences (*Accept-Language*, *Accept-Encoding*, *Accept-Charset* headers), the URL of the Web site the user has visited previously (*Referer* header) and the user's mail address (*From* header). There are several Web sites where we can see the headers our browser is being sent in a HTTP request, e.g., (Langton, 2011; Gemal, 2011).

A cookie is a mechanism defined in order to come up with state in HTTP (Barth, 2011), which allows carrying out multistep transactions. Hence, cookies are essential for the support of shopping carts, personalization based on user's preferences, identification of an authentication session, automatic login, location memory, customer classification, etc (Harding et al., 2007; CISCO Systems, 2009; Yue and Wang, 2009).

Namely, a cookie is a string of text that contains a name and its value, an expiration date (established in the optional *Expire* and *Max-Age* attributes) and the originating site (established in the optional *Domain* and *path* attributes) (Barth, 2011).

Cookies can be established for the domain the user is downloading the Web page (known as *first-party cookie*, in Figure 1 in steps 2.1 and 2.2) or for another different domain (known as*third-party cookie* in Figure 1 in steps 2.3 and 2.4).

A cookie is sent by the Web server in the *Set-Cookie* header and the server can send several cookies by incluiding in the same response as many *Set-Cookie* headers as cookies to be established. Once the cookie is established, the Web browser will send it each time user accesses the Web. The cookie is sent by means of the *Cookie* header.

Depending on the lifetime of a cookie, they are classified as session cookies and persistent cookies. The former are those that are erased when the Web browser is closed. Thus, this cookie is not stored in the user's hard disk, only resides in memory and the risk they produce is quite reduced. The latter are stored in hard disk and persist even when the Web browser is closed (or until they expire or user deletes them). For this reason they are also named *tracking cookies*.

Cookie mechanism has been used broadly to track, profile and monitor user's browsing activities (Rezgui et al., 2003; Senicar et al., 2003; Linn, 2005; Yue et al., 2010; Barth, 2011). Moreover, they could be manipulated or stolen (Yue et al., 2010).

For tracking purpose, a cookie (or some of them) could contain the number of times the user has visited the Web and the Web pages you have visited and when you have visited them. Even it can store user's movement in the Web site. Furthermore, cookies can be combined with other information obtained from the HTTP headers (e.g. *Referer* header) and with Web bugs (see next section) to obtain more precise information on a user. Due to its implications related to privacy, even some legislation have appeared in both Europe and United States (Miyazaki, 2008).

In order to see the cookies that are sent and received when we access a Web site we can use several tools such as (Odvarko, 2011). More information on cookies can be found in (Cookies.org, 2011).

2.2.3. Application layer

In this layer we consider a set of technologies that are on top of HTML as well as Web applications that do not request explicitely personally identifiable information (that is, we do not need to be registered and authenticated in that applications) since if user is authenticated all the information is available. The analysis of privacy in this scenario requires the analysis of privacy identity solutions, which is out of the scope of this paper.

Namely, apart from HTML tags, we are referring to objects that are embedded in Web pages such as Web bugs (Martin et al., 2003), banner ads, pop-up and pop-under windows, JavaScript, VBScript, ActiveX, Java applets, Flash objects and plugins. Some of these objects are active objects whose purpose is to improve interactivity and the incorporation of multimedia content in HTML. Thus, they improve user's experience and enhance user interfaces. Some of these objects, apart from being a potencial source of PII leakage, decrease download performance (e.g. pop-ups, banner ads, etc).

As for Web applications that do not explicitely collect data but could create profiles we refer to Web search engines since they can obtain identifying information such as user's name, social security name, location, user's work, family, interests and future plans (Saint-Jean et al., 2007; Castella-Roca et al., 2009; Peddinti and Saxena, 2010).

A *Web bug* (also known as Web beacon, 1x1 gif or tracking bug) is an element or object (generally it is a transparent GIF) that is embedded in a Web page or even in an e-mail (Martin et al., 2003). The purpose of including this element is to know whether the user has viewed the Web page or e-mail where it has been embedded.

In general, Web bugs are used by third parties to monitor user activity (count unique and repited visitors as well as how they have entered the Web site) or to elaborate statistics. Third party entities (advertisers, DoubleClick, Google analytics, 2mdn.net, etc) can know which entity made the request from the *Referer* HTTP header or by using dynamic URLs (Barth, 2011). In the Figure 1, the obtaining of a Web bug would be depicted in steps 2.3 and 2.4.

Web bugs can also be combined with cookies (third-party cookies) in order to know the computer the user is accessing (they also use the *Referer* header in order to know from which site the user comes from), the Web page opened, when the visit started, the number of times the user has accessed to that third party Web site and from which servers she has accessed (this Web bug could be placed in the Web site of different companies) (Martin et al., 2003; Harding et al., 2007; Miyazaki, 2008). Thus, a third party can track a user across multiple Web sites and create the profile of that user. This technique is each time more used in Web sites during the latest years (Miyazaki, 2008). A report on the sites with most Web bugs and its tracker coverage can be found in (KnowPrivacy, 2011).

Javascripts, ActiveXs, Java applets, Flash objects and plugins can represent a privacy threat since they can be used to fingerprint the user's machine and thus, identify the user (this could be made even without cookies, although with its use could provide better results) (Martin and Schulman, 2002; Saint-Jean et al., 2007; Eckersley, 2010).

Basically, the fingerprint is the identification of a set of browser features such as user agent, content-types of the HTTP *Accept* header, screen resolution, timezone, brower plugins, plugins versions and MIME types, systems fonts and some information provided by some tests for cookies (Eckersley, 2010). If this information is distintive enough, it allows the identification of a user.

By means of JavaScript we can obtain different information of the Web page when it is executed in user's machine such as Web page information, cookies and location bar. This allows that different attacks such as cookie stealing, location hijacking, history sniffing and behaviour tracking (Jang et al., 2010) can be carried out. The user could be fingerprinted even by her typing (Chairunnanda et al., 2011).

Furthermore, some components such as Flash Objects also handle cookies (known as Flash cookies or local shared objects) (Krishnamurthy and Wills, 2009) that can help to track users and obtain information from the user such as computer's configuration or information to provide to the Web site. In BrowserSPY (Microsoft Corporation, 2011) you can check whether your Flash is enabled to store cookies.

As for Web applications that do not use PII and that are commonly accessed we can point out Web search engines. They can obtain, from the request information contained in the previous layers, information such as IP and HTTP headers. Web search engines can also carry out inference and linkage on query terms, redirects in the results provided and Web timing attacks (cache timing attacks) to distinguish among users (Jackson et al., 2006; Saint-Jean et al., 2007). In fact, for some activities such as behavioral targeting, the information provided by search queries is several times better than information provided by the pages that user clicked (Yan et al., 2009).

This information combined with the time of day (as well as on-line information) let the Web search engines obtain valuable information about the user (user's work, interest, future plans, etc) and her activities at a specific time (Saint-Jean et al., 2007).

In (Gemal, 2011) you can find a Web site that shows what information can be retrieved from your browser based both on the Web objects mentioned in this layer as well as in the previous layer.

3. Solutions and tools for private navigation

In this section we mention the different solutions that have been proposed in order to cope with the privacy problems stated for each layer in the previous section.

Once the solutions have been introduced in Sections 3.1, 3.2 and 3.3 we present in Section 3.4 the different tools that are freely available and that implement these solutions. It is important to highlight that in order to offer a comprehensive solution to privacy on the Web we should combine the use of the solutions presented in each level.

3.1. Privacy solutions for TCP/IP layer

The solutions that offer privacy at TCP/IP layer are usually known as systems or solutions for anonymous communications. The aim of these solutions is to offer protection against traffic analysis since this can be used either to obtain information on identification or for profiling or for information extraction.

An ideal system for anonymous communications should be prevent the following types of attacks (Berthold et al., 2000; Back et al., 2001; Raymond, 2001): message coding, timing, message volume, flooding, intersection, collusion and tagging.

The most simple solution proposed for this purpose is to use encryption by means of SSL/TLS, Virtual Private Networks (VPNs) or tcpcrypt (Mazieres et al., 2011). However, neither of these solutions prevent that parties that are exchanging information can be identified. Even though the communication is ciphered, the content being accessed could be determined from the size data exchanged, the time and frecuency of the communication or statistics of information exchanged (traffic signatures) (Hintz, 2002; Sun et al., 2002; Bissias et al., 2005; Liberatore and Levine, 2006; Danezis and Clayton, 2007; Herrmann et al., 2009). Encryption do not protect either of privacy compromises that can be performed in the Web sites user accesses (Linn, 2005; Danezis and Diaz, 2008).

The level of protection can be increased with the use of a Web proxy, which is a (trusted or semi-trusted) entity that receives user's Web requests and made them on behalf of the user. Thus, the Web proxy hides the IP information of the user that is originating the request. However, the trace of all the traffic that is originated in and have as destination the proxy might reveal user's identity even SSL/TLS is used (Gabber et al., 1999; Danezis and Diaz, 2008; Li et al., 2011). Furthermore, the Web proxy knows all the information and can trace user's activity (Margasiński and Szczypiorski, 2005). A proxy cannot either prevent that the user's Internet Service Provider (ISP) monitors her activities if a SSL/TLS connection is not used in the connection with the Web proxy. On the other hand, proxy main advantage is that it is a low lacenty system (Edman and Yener, 2009).

In this layer, the best level of protection can be achieved by means of solutions that are based on the use of chains of specials proxies that send and receive information in an encrypted way and that do not know the Web server where the information is requested. In general, the only information that this kind of intermediary proxies (hereinafter we will name them as anonymous routers independently if the follow the concept of Chaum's Mix or onion or garlic routing) knows is its precedessor and succesor in the chain. With these chains, user's ISP can only see that you are connected to an intermediary proxy and the Web sites only see that they receive request from these anonymous routers. These solutions also offer protection against traffic analysis. We can distinguish different kind of solutions that follow this approach and that can be classified in four main groups (Ren and Wu, 2010): Mixnet-based schemes, DC-net systems, network routing-based techniques and peer-to-peer networks.

Although there are an important number of proposals in each group, see (Danezis et al., 2009; Ren and Wu, 2010) for more proposed techniques, in the following subsections, only the description of the techniques that are implemented in the tools commented below is provided.

Our goal is to analyse only those solutions that can be used in a practical way for end users (without any special technical knowledge) as a solution for privacy problem in the Web environment. As we will see, the proposals that have a development freely available for Web users is quite reduced. Namely, the techniques that we describe are: Tor (Dingledine et al., 2004), JAP/JonDo (Web Mixes) (Berthold et al., 2001) and I2P (zzz and Schimmer, 2009). It is important to point out that these solutions need be combined with solutions of other levels since they only anonymize TCP/IP level.

Otherwise, the user could be identified by means of some of the techniques that we have outlined in the previous section and that will be explained in more detail in the HTTP or/and application layers.

Other approach could be not to provide anonymity in the transport layer and implement it in the application layers. However, as mentioned by Berthold et al. (2000), this approach is less suitable and the privacy with solutions at that level could be hardly obtained. Therefore, ideally, any privacy solution should be based on the provision of privacy at transport level. This also has additional advantages (Berthold et al., 2000): the imposibility of distinguish between the use of different kind of services and the freedom, for the different parties, to decide whether they want to reveal their identity although they use a privacy solution at transport level.

Next we present a description of the different solutions that help in providing privacy at the TCP/IP layer.

3.1.1. Tor

Tor (Dingledine et al., 2004) is an improvement of the onion routing proposal (Goldschlag et al., 1996; Reed et al., 1998; Syverson et al., 2001). In fact, it was presented as the second generation of onion routing.

Tor is a distributed overlay network for providing anonymous communications. Tor is based on the establisment of a virtual circuit using an incremental or telescoping path-building unlike onion routing where a onion structure is used. Furthermore, Tor provides perfect forward secrecy, congestion control, directory servers and location-hidden services.

The main elements of this proposal are: onion proxies, onion routers and directories servers. These elements are described next.

An *onion proxy* is executed in the local machine of each user. This proxy is responsible for building circuits across the network using the information provided by directory servers. The building of the circuit is performed hop to hop negotiating a symmetric key with each onion routing that will be part of the circuit. This proxy also handles the requests of the user's applications and sends them through the circuits established. As interface for applications SOCKS is used and for the maintaining of privacy features they propose the filtering with an application-level proxy such Privoxy or Polipo. The information is sent in structures of fixed size (512 bytes) named *cells*. Each cell is ciphered with each one of the keys negotiated with the onion routers in the building path. For the recipient the traffic seems to be originated from the exit onion router. The traffic between exit and destination is not ciphered.

An *onion router* is a node in the overlay network in charge of relaying information from/across other onion routers and proxies. Onion routers are connected one another and with onion proxies by means of a TLS connection.

Finally, *directory servers* maintain the list of onion routers available, the status of network topology and the keys and exit policies of each onion router.

Tor is a low-latency system (Edman and Yener, 2009) that guarantees perfect forward secrecy and sender anonymity. Moreover, recipient anonimity is guaranteed when location-hidden services are established.

Currently, Tor is the most used anonymity system (Li et al., 2011). Tor is mainly used for HTTP, BitTorrent and SSL (Mccoy et al., 2008; Chaabane et al., 2010). As for the most Web categories visited are search engines, pornography and computers and Internet (Chaabane et al., 2010). An study on the latency of this system can be found in (Wendolsky et al., 2007; Fabian et al., 2010).

Furthemore, Tor has been defined as the anonymity layer in Privacy and Identity Management for Europe (PRIME) project (Ardagna et al., 2010), which is an European project with the aim of providing a Privacy-enhancing Identity Management environment which covers both technical and non-technical (legal, social and economic) issues. Namely, its aim is to offer real communication solutions for users in information society while interact in a safe way and retaining the control of their privacy.

More details and analysis of Tor can be found in a wide number of references in the literature since this protocol has been deeply analysed (Murdoch and Danezis, 2005; Abou-Tair et al., 2009; Danezis et al., 2009; Behl and Lilien, 2009; Edman and Yener, 2009; Chaabane et al., 2010; Fabian et al., 2010; Ren and Wu, 2010; Mulazzani et al., 2010; Hopper et al., 2010).

Even it is the largest deployed anonymity network (Edman and Yener, 2009), several attacks have been found, some of these are due to the fact that Tor is not designed to provide security against even passive observers of a circuit (Danezis et al., 2009) and it does not offer protection at the boundaries of the network (Serjantov and Sewell, 2003; Murdoch and Zieliński, 2007).

Murdoch and Danezis (Murdoch and Danezis, 2005) have shown as a malicious Tor node can determine the nodes of a Tor circuit (timing-based attack). Hopper et al (Hopper et al., 2007, 2010) have found out two attacks. The first attacks allows, by means of a pair of colluding Web sites to determine (with high confidence) whether two connections that make use of the same Tor exit node are using the same virtual circuit. The second allows a corrupt Web site to obtain several bits of information of each access the user makes. A more detailed description of other attacks can be found in (Hopper et al., 2007; Snader and Borisov, 2008; Edman and Yener, 2009; Danezis et al., 2009; Hopper et al., 2010).

3.1.2. Web MIXes/AN.ON project

Web MIXes (Berthold et al., 2001) is a system designed to provide anonymous communications for both asynchronous and synchronous traffic. This system, which was developed in the AN.ON project (Berthold et al., 2000; Golembiewski et al., 2003; AN.ON Project, 2011), is based on the modification on Chaum Mix concept (Chaum, 1981).

This solution is built on four components that are used to build an anonymous tunnel (Berthold et al., 2001; Golembiewski et al., 2003): Java Anon Proxy (JAP), MIXes, cache-proxy and Info-service.

JAP is a program installed on user's computer and is used to send anonymous traffic through the *MIXes*, the MIXes are a set of servers that follow the idea of MIX server proposed by Chaum, the information on them is obtained through the info-service, and finally, cache-proxy sends and receives the traffic from the (Web) servers. Next, we point out the main features of each of these components.

JAP is a proxy that applications in user's computer use to send anonymous traffic. The traffic is sent to the MIXes periodically in slices of a fixed size by using an adaptive chop-and-slice algorithm. If traffic is not generated by applications, then, dummy messages are sent. It is important to point out that JAP does not also prevent

from leakage of PII in this layer but also performs the filtering of elements of upper layers that can comprise user's privacy such as cookies, JavaScript, etc. Theses issues will be described in more detail below.

MIXes are based on the idea of Chaum mix cascade (change of cryptographic coding, re-order and mix messages received and send them in a batch), each message goes through all the cascade of mixes in the same order and they also generate dummy traffic when they do not have "real" traffic to send. The traffic in a MIX is received from the JAP and the exit point of a MIX is a cache-proxy.

Cache-proxy is a reverse proxy from (Web) servers that receives requests from the MIX. It also generates dummy traffic when there are no requests. Furthermore, we can point out that this proxy returns all the Web page requested with the objects that are embedded in it.

Info-service is in charge of management tasks in the system. Namely, it provides information on the MIXes: addresses, public keys, availability, the traffic situation and the level of anonymity (number of active users in the system) (Pfitzmann and Hansen, 2010).

This solution also defines a ticket-based authentication system that prevents flooding attacks. Currently, this solution is also known simply as JAP or JonDo. This latter name is used for the comercial version provided by JonDonym anonymous proxy servers (AN.ON Project, 2011).

More details and analysis of Web MIXes can be found in (Danezis et al., 2009; Edman and Yener, 2009; Ren and Wu, 2010; Westermann et al., 2010; Westermann and Kesdogan, 2011).

Although this solution in its design proposes the sending of dummy traffic, in the implementation is not used due to the load that would suppose for the network (Edman and Yener, 2009). This fact limitates the level of protection offered by the solution. The messages in this solution could also be tagged in order to recognize them when they are decrypted (Danezis et al., 2009).

In (Westermann et al., 2010) we can find out two main flaws related to the session key used in the mixes, which are not checked if they are fresh enough and thus a replay attack could be made, and the encryption scheme used, which can cause the de-anonimization of the users. Other attacks that are based on replay, which can disclose some of the visited Webs by the user can be found in (Westermann and Kesdogan, 2011).

3.1.3. I2P

I2P (zzz and Schimmer, 2009), which is an evolution of Invisible Internet Project, is defined for providing secure and anonymous communications both the sender and the receiver.

I2P is based on garlic routing instead of onion routing. Thus, not only the communication between routers is ciphered but also end-to-end communications, allowing at the same time sending multiple messages in the same layer of protection.

According to I2P terminology we can distinguish three main elements: tunnels, routers and network database (Kubieziel, 2007; zzz and Schimmer, 2009).

Tunnels are established in order to send information anonymously. In I2P there are different tunnels for incoming and outcoming traffic: inbound and outbound tunnels. In order to establish a communication between two peers, the creator of the tunnel sends the information by using her outbound tunnel, when the traffic arrives at its last router in the tunnel (named as *outbound endpoint*), this endpoint sends the information to the inbound tunnel of the receiver. The first element in this tunnel is named as *inbound gateway* as is responsible for relay the traffic to the destination. The tunnels are built up from peers that are chosen randomly after being classified in tiers. The classification is based on capacity, latency and whether the peers are overloaded.

Routers are the elements participating in the network to relay traffic from senders to the destination. It is important to point out that the destination (named *Eepsites* for Web pages) in this solution is always anonymous.

The *network database (netDb)* contains the information that allows the location of elements available in the network. Namely, this network database manages network metadata that allows peers to know information to send traffic to a router (*routerInfo*) as well as how to locate a particular destination (*leaseSets*).

More details and analysis on I2P can be found in (Abou-Tair et al., 2009; zzz and Schimmer, 2009; I2P, 2011; Herrmann and Grothoff, 2011; Zantout and Haraty, 2011). Although this system offers protection against a number of attacks such as timing attacks, intersection attacks, taggin attacks, sybil attacks, etc, it presents some possible vulnerabilities as for partitioning attacks and intersection attacks (Zantout and Haraty, 2011), which could reveal sender and receiver identities or allow the trace of the message. Herrmann and Grothoff (2011) shows and attack based on taking over the fast tier in order to identify the peer hosting an Eepsite.

3.2. Privacy solutions for HTTP layer

In the HTTP level, there are two main elements to take into account in order to protect user's privacy: HTTP

headers and cookies. The solutions considered to overcome these problems are: the filtering of headers that can reveal (or induct) some PII, and the filtering, blocking or limitation on the use of cookies.

For the filtering of HTTP headers, there are two approaches. On the one hand, the installation of Web browser components/plug-ins or local proxies in user's computer. On the other hand, the use of anonymous Web proxies or anonymizers that perform this task. The functions of the elements of each approach are commented below.

As for cookies, there are several options to prevent they can be used to track user's activity. First, disabling them in the Web browser. However, this option is not suitable because can cause usability problems since many Web sites, for some transactions (shopping carts, payment transactions, etc), that use them. Second, deleting manually cookies at the end of each browser session (Harding et al., 2007). Third, the use of tools or plug-ins for the management of cookies (Shankar and Karlof, 2006; Yue et al., 2010). Four, the use of anonymous Web proxies (on the user's computer or in a server). Finally, private browsing mode implemented by Web browsers. The different elements used in these options are commented below.

Next, we provide a description of the different solutions that help in providing privacy at HTTP layer.

3.2.1. Cookie manager

A cookie manager is a tool that allows the management of cookies: view, edit, establish, filtering options, delete them, etc. Generally, many of these functions are included as part of the functionality of the Web browsers.

In (Cookies.org, 2011) you can find information on how to manage them in the different browsers. However, in many cases, the functionality offered by Web browsers is quite limited and the user can improve this functionality by installing add-ons or extensions for the Web browser that complements its functionality.

With this kind of tool the user could limit the number of cookies accepted and remove or filter those that could be used to track the user.

3.2.2. HTTP filter

An *HTTP filter* is in charge of modifying the HTTP request the Web brower sends to the Web server and erases or modifies those headers that can reveal PII (user's agent, referrer, etc).

For those cases that headers cannot be removed, the use of generic values that cannot fingerprint the user (Eckersley, 2010) is proposed, e.g., the *Accept-Language* header is used to indicate the language the user's accepts, if the language is from a very particular region, the user could be identified. Thus, for this header the proposal is to indicate English as language. With this configuration is more difficult to induct information from the user who is making the request since there are many users who work with that language in her browser.

The same situation could happen with other headers such as the user's Web browser and its version (*User_agent* header), if the version is very specific, then the use of a common version is recommended to be replaced. Therefore, as mentioned in (Saint-Jean et al., 2007), its aim is to normalize the HTTP request. Most of the times, this functionality is included as a part of an anonymous Web proxy.

An analysis on how specific (unique) and trackable is your browser and the bits of information can be obtained is shown in (Eckersley, 2010; Electronic Frontier Foundation, 2011). This information is based on the following values: user agent and HTTP_ACCEPT headers, browser plugin details, time zone, screen size and color depth, systems fonts, if the cookies are enabled or not, and a test to determine if supercookie is limited. A cookie is named as supercookie when the domain is a public suffix domain (e.g., .org, .com, .co.uk, etc).

3.2.3. Simple anonymous Web proxy

An *anonymous Web proxy* (also known as an *anonymizer*) acts as a TCP proxy and removes headers with user's information (or fake them), conceals user's IP address (Shubina and Smith, 2003) and rewrites HTML pages so that when the user clicks on a link on that page, the request is made through the proxy. In general, it also manages cookies on behalf the user. Additionally, some of them also remove active contents (Javascripts, banners, advertisements, etc) and other embedded objects from the HTML. However, this issue will be covered later in the next layer.

In this level we will suppose that the filtering of active content is not performed or the proxy does not support this feature.

In order to distinguish the features offered by an anonymous Web proxy in this level and with another with more advanced features for the following level, we will name them as *simple anonymous Web proxy* and *(Advanced) anonymous Web proxy*. In this section, we analyse the former. The latter is analyses subsequently in Section 3.3.2.

A simple anonymous Web proxy in user's local computer can remove some HTTP headers (as an HTTP filter) and can manage cookies on behalf the user but, at

the end, the request comes from the same IP address and therefore, the IP address of the user can be identified.

If the proxy is in a third party, the IP address of the user cannot be identified. Even though, if no additional protection measures are taken, the user can be traced whether the user's traffic is traced (even it is protected with SSL/TLS) (Gabber et al., 1999; Danezis and Diaz, 2008; Edman and Yener, 2009; Li et al., 2011). But even though, the user's identity may be revealed if all communications to and from the proxy are traced. Furthermore, this solution does not prevent that both the Web proxy and the user's Internet Service Provider (ISP) trace all her Web activities (Margasiński and Szczypiorski, 2003, 2005). Neither the ISP nor an eveasdropper could obtain information on the connection if user uses SSL/TLS to connect to the Web of the anonymous Web proxy.

Its advantages are the high efficiency they can offer, easy to access and to use, simplicity and do not require additional elements (Margasiński and Szczypiorski, 2005; Edman and Yener, 2009).

The main disadvantage is that a simple anonymizer does not protect against traffic analysis even though a SSL/TLS connection is being used (Hintz, 2002; Sun et al., 2002; Margasiński and Szczypiorski, 2003; Bissias et al., 2005; Margasiński and Szczypiorski, 2005; Liberatore and Levine, 2006; Danezis and Clayton, 2007). It does not batch and reorder messages either (Edman and Yener, 2009). If the proxy only covers headers and the additional objects that are embedded in the HTML are not processed, then the user's privacy can be compromised (Margasiński and Szczypiorski, 2003) by means of the information provided in the application level.

3.2.4. Private browsing mode

The private browsing mode is a feature that have been included recently in Web browsers. In Mozilla Firefox and Safari is named *Private Browsing*, in Microsoft Internet Explorer *InPrivate* and in Google Chrome *Incognito*. This mode of navigation aims not to leave trace on the user's computers on the Web sites she has visited and hides the identity of user from the Web sites she visits (Aggarwal et al., 2010).

Basically, private browsing mode is based on not to store some information after the private navigation session has finished and not make it available in the public mode of navegation as well as it is responsible for disabling toolbars and extensions since they can compromise user's privacy.

Mainly, the information considered is the browser navegation history, the cookies of the session, password database, cache of the Web browser, client's certificates. However, how this mode is implemented depends on the browser (Aggarwal et al., 2010). Thus, this solution can be basically used to limit the effect on the use of cookies and the information that can be obtained with active components, e.g., it can prevent the access to history by means of Javascript (more details are provided in the next layer).

3.2.5. Do Not Track

Do Not Track (DNT) (Mayer et al., 2011) is a recent technology that aims to improve user's control on the PII is released to third party entities when user accesses to a Web site. Namely, with this proposal the user when accesses to a Web site she indicates that she does not want to be tracked by third parties (including behavioral advertising). This is indication is made by means a new HTTP header (*DNT*) that the user's Web browser sends to the Web site.

The support of this technology is not mandatory by Web sites and it needs to be accompanied with some legislation that requires (enforces) its compliance.

Currently, DNT has been submitted as an Internet-Draft to the IETF (Mayer et al., 2011) in order to become an standard. As it is a recent technology almost is not supported by most Web sites. The DoNotTrack.Us Web site (Mayer and Narayanan, 2011) allows us to check whether our browser supports this extension and whether it is enabled.

3.3. Privacy solutions for application layer

In this level the elements that can compromise user's privacy are the elements that can be embedded in a Web page through HTML tags or as objects such as Web bugs, banners, advertisements, JavaScript, ActiveX, Java and other posible plugins (Silverligth, etc) since they can be used to send PII to the Web server or to fingerprint user's machine and, therefore, identify user (Saint-Jean et al., 2007).

The use of Web search engines can also compromise user's privacy since Web search engines can profile user in function of the queries the user makes (Saint-Jean et al., 2007).

In general, the solution to these problems introduced by Web objects is to disable or block them at Web browser (Eckersley, 2010). Thus, these elements will be not loaded, executed and displayed when the user loads a Web page and how they will not be executed they will not cause any PII leakage. However, this is not suitable since it causes usability problems. For these elements different solutions have been proposed and we mention them next.

In order to avoid privacy risks produced by Web bugs the solution is to analyse the content of the HTML page received in an HTTP request with the aim of filtering the request of very slow size images (7 pixels or less) different from the Web server where the user made the request. This task can be performed by a HTML filter or an advanced anonymous Web proxy.

There are several proposals in order to detect a Web bug (Alsaid and Martin, 2003; Martin et al., 2003; Fonseca et al., 2005; Ragkhitwetsagul, 2007; Yamada et al., 2010, 2011; Baviskar and Thilagam, 2011). These proposals take into account the image domain, the size of the image, if the image has a third-party cookie, if the URL of the image contains more than a protocol, the length of the image URL and link analysis. Even the use of blacklists has been proposed. However, this later mechanism is not suitable since they are generated by Web crawlers or by volunteers (Yamada et al., 2010, 2011). A complementary solution of backlists based on temporal link analysis is proposed in (Yamada et al., 2011). This task can be performed by a HTML filter or an advanced anonymous Web proxy.

As for JavaScript several techniques have been proposed (Nentwich et al., 2007; Yu et al., 2007; Jim et al., 2007; Dhawan and Ganapathy, 2009; Chudnov and Naumann, 2010; Jang et al., 2010) such as solutions based on client-side or server-side to prevent history sniffing, disable unknown scripts, signed scripts, program instrumentation and dynamic taint propagation and checking. Some of these solutions can be implemented in proxies and other requires the modification of Web browser source code.

In order to prevent Web search engines can profile user, she can use private Web search tools.

Next, we provide a description of the different solutions that help in providing privacy at application layer.

3.3.1. HTML filter

An HTML filter is in charge of removing any Web object (JavaScripts, Flash, Java applets, ActiveXs, popups, etc) that can provide user's PII at the same time that it carries out only one HTTP GET request per Web access (Aggarwal et al., 2010).

Depending on the filter, different options could be offered: remove all Web objects of a particular kind, only for specific Webs, etc. This functionality is also incorporated by some anonymous Web proxies.

In general, the blocking of all Web objects can also be configured by means of the configuration options of the Web browser. However, it can cause usability problems if user wants different kind of accesses.

3.3.2. Advanced anonymous Web proxy

In this section we include the simple anonymous Web proxies that satisfy the features mentioned in Section 3.2.3 (that is, filtering HTTP headers and cookies) as well as those that incorporate the HTML filter feature, we have just mentioned in the previous section. Thus, when the advanced anonymous Web proxy receives the Web page requested by the user from the Web server, it parses it and removes any Web object that can compromise user's privacy (Margasiński and Szczypiorski, 2005; Saint-Jean et al., 2007).

As for advantages and disadvantages of these systems, we can mention that they share the same features as simple anonymous Web proxy (Section 3.2.3) and improve their features avoiding privacy compromise by Web objects. Even though, as already explained, the user could be identified by means of traffic analysis.

3.3.3. Private Web search tools

A private Web search tool aims to prevent that a Web search engine such as Google, Bing, etc can build a profile of the user from the queries she makes.

There are two kind of private Web search tools: Private Web search engines and Private Web search plugins. The former are Web search engines that act as a proxy between the user and a well-known search engine. These private Web search engines delete the cookies that Web search engines uses to track the user as well as the identifier assigned to each user. The latter are tools that implement some tool to prevent the query the user makes to the Web search engine cannot be profiled. In general, this kind of solutions are based on the obfuscation of the real query between other queries that are randomly genarated.

3.4. Tools

In this section we present the main tools that are freely available to cope with privacy issues and that develop some of the solutions mentioned in the previous sections (Sections 3.1 to 3.3).

For each tool analysed we provide a brief description with its features, the different solutions that implements and whether it should be complemented or not with the use of other tools that cover privacy in the different levels already presented.

3.4.1. Multiproxy

Multiproxy (Multiproxy, 2001) is a tool that is installed in user's computer acting as a TCP proxy. Each time Multiproxy receives a request, it redirects it to a different proxy server from a list of proxy servers.

This solution conceals user's IP address using different proxy servers (the choice of the server is in function of its speed). Thus, the access is more reliable (your Web request does not depend on a single server that could fail). At the same time, it is more difficult for a Web site to determine the IP address. Furthermore, in order to track the user, an attacker has to eavesdrop to more proxies. However, this solution has the problems already introduced in Section 3.1. Thus, the user could be identified by using information at HTTP and application layers. Moreover, the ISP could analyse user's requests and behaviour.

The proxy servers to be used with this tool can be obtained from one of the numerous public list available on the Web. Some list can be found in (Rosinstrument.com, 2011; PublicProxyServers.com, 2011; MyProxy, 2011). This proxy servers can be also used to configure our Web browser with a proxy.

3.4.2. CGIproxy

CGIproxy (Marshall, 2008) is a CGI proxy written in Perl. Therefore, it could be executed in different platforms.

This proxy supports the filtering of HTTP headers, management of cookies and the removing of Web objects. It allows the Web administrator to configure an important number of options related to these issues. Thus, we could configure an advanced Web anonymous proxy (see Section 3.3.2). Although this solution covers several levels, user privacy can be compromised by means of traffic analysis.

3.4.3. Privoxy

Privoxy (Privoxy, 2010) is a non-caching filtering proxy that supports both IPv4 and IPv6 and incorporates filtering capabilities. Furthermore, Privoxy supports SOCKS protocol, the filtering of HTTP headers, the management of cookies and it removes Web objects (Web bugs, banners, advertisments, Javascript, etc) that can compromise user's privacy. Therefore, it can behave as an advanced anonymous Web proxy, which does not prevent traffic analysis attacks.

3.4.4. Polipo

Polipo is a caching Web proxy (Chroboczek, 2010) that supports HTTP/1.1 both for IPv4 and IPv6. It stands out because of its support of HTTP/1.1 pipelining as well as Poor Man's Multiplexing to reduce communication latency.

Due to the fact that it supports the SOCKS protocol is being used with the Tor anonymizing network. In fact, its use is recommended with the Tor browser bundle (which will be explained below) in order to improve Tor's communication latency. Furthermore, it supports the filtering of HTTP headers and cookies as well as blocking of Web bugs or advertisments by blocking or redirecting URLs (content filtering).

Both Privoxy and Polipo can be used together with Tor bundle, although it seems that Polipo is better for this use due to its support of pipelining (TOR FAQ, 2011). Furthermore, there are some Graphical User Interfaces (GUI) for this tool: Solipo (Solipo, 2010), for Windows and Dolipo (Dolipo, 2008), for MAC OS X.

3.4.5. Tor

Tor (The Tor project, Inc, 2011a) is the implementation of the solution presented in Section 3.1.1. This tool, as is, is only to be used by expert users. To end users it is recommended to use, at least, Vidalia or Tor browser bundles.

Vidalia is a GUI to control Tor, that is, it allows users to decide when they want to be connected/disconnected to the Tor network, see the bandwidth used, the active circuits, Tor's current state and configure a Tor client, bridge or relay. The Tor browser bundle is commented below.

With Tor, even with the use of Vidalia (The Tor project, Inc, 2011d), we can only achieve privacy at TCP/IP layer. However, privacy can be compromised using elements of HTTP and application layers since if we have only installed Tor, the Web browser connects directly, by means of SOCKS to the Tor network. This is the reason why Tor recommends the use of a proxy such as Polipo or Privoxy. In fact, as we will see below, the Tor browser bundle incorporates Polipo. This Web proxy behaves as an advanced anonymous Web proxy, which also handles elements of the other levels.

3.4.6. Torbutton

Torbutton (The Tor project, Inc, 2011c) is an add-on for Mozilla Firefox to work together Tor (it has to be previously installed). This tool allows enabling and disabling Tor with only one click in Firefox and it disables Web objects and active content (such as Javascript, Flash objects, etc) that can be incorporated in Web pages. Furthermore, it supports the configuration of other features that can compromise user's privacy such as disabling search suggestions from Google, blocking the indication whether some links have been visited, prevention of storing history of visited URLs and password forms, blocking disk and memory cache, management of cookies, management of headers related to the language so

that it appears as an English browser and prevention of sending the *Referer* header.

Hence, this tool (with Tor) could provide anonymous Web browsing considering all the privacy concerns presented in the different levels. However, this combination has a flaw: the DNS requests that the browser performs are not made through the Tor network, they are made via the user's computer. Thus, an attacker by means of DNS traffic analysis can know the names (and its domain) the user is visiting throught Tor network. For this reason, the use of Polipo or Privoxy is recommended. Indeed, Polipo is included in the Tor browser bundle as we explain in next section.

The usability of this tool has been studied in (Clark et al., 2007), where it is concluded that this tool is easy to configure, install and use as long as Privoxy/Polipo is used. They also mention that with this tool, to enable and disable Tor is more intuitive. In the event Privoxy is not used, it requires some improvements as for the configuration steps. In this study they also mention that it provides a better interface than FoxyProxy, which is commented next.

3.4.7. FoxyProxy

FoxyProxy (Jung, 2011) is an add-on for Mozilla Firefox (it will be soon for Google Chrome and Microsoft Internet Explorer), which allows the definition of the proxy to use in function of the URL patterns chosen by the user.

Clark et al. (2007) mention that to enable and disable Tor is more intuitive with this tool than with Torbutton. Furthermore, we can indicate easily that all the traffic goes through Tor, which solves the DNS problem mentioned with Torbutton. However, FoxyProxy does not provide any functionality related to the solutions mentioned for HTTP and applications layers. Thus, if FoxyProxy is combined with Tor we obtain privacy at TCP/IP level but not for the other levels. In order to obtain more privacy we should combine it with Polipo or Privoxy.

3.4.8. UnPlug

UnPlug (Dbatley, 2011) is an extension to download Flash videos. The main feature of this tool is that the video is downloaded in user's computer before playing it. Thus, it avoids the activation of Flash in the Web browser and, at the same time, it improves performance since additional reproductions do not require a connection to the Web server. Hence, this solution helps preventing some problem at the application layer, particularly, as for active content in Flash.

3.4.9. Plugin customs

Plugin customs (Startingpage, 2011) is an extension for Safari that allows the blocking of different plug-ins such as Flash, Silverlight, Java, etc. It is important to point out that it supports the customization on the Web sites of the plug-in can be used to show specific resources. Thus, this application works at application level with active objects.

3.4.10. Tor browser bundle/Vidalia bundle

The Tor browser bundle (The Tor project, Inc, 2011b) contains Tor, Vidalia, Polipo and Mozilla Firefox Portable (a modified version of Mozilla Firefox to make it portable and that does not leave personal information in your computer) with Torbutton installed (see Section 3.4.6).

The Vidalia bundle is practically the same as Tor browser bundle except Mozilla Firefox is not contained in the bundle and the user needs to have installed it previously.

Tor browser bundle is the recommended option for end users since it installs a set of components that are needed for protecting privacy of Web communications. This bundle provides protection against almost all the privacy concerns presented in the three layers. Thus, the user has a comprehensive solution for her private navigation.

The privacy risks associated to this solution are mainly those described when we introduced Tor network (see Section 3.1.1) and those derived of using a Web search engine without a private Web search tool.

The usability of this tool has been analysed in (Clark et al., 2007; Abou-Tair et al., 2009; Schomburg, 2009) and the authors of these works conclude that the tools provided with the bundle are easy to use, although some issues in the installation and configuration should be improved (mainly for facilitating its use for novice users). As mentioned by Edman and Yener (2009), this bundle might have contributed to popularity of Tor.

Currently, Tor network is the largest anonymity network (with 10387 servers) and the most used (Li et al., 2011).

Furthermore, the performance of this network has been studied in (Wendolsky et al., 2007; Panchenko et al., 2008; Abou-Tair et al., 2009; Loesing et al., 2008; Lenhard et al., 2009; Fabian et al., 2010).

Fabian et al. (2010) and Panchenko et al. (2008) mention that the latency should be reduced so that the adoption of Tor network service by new users increases. As mentioned in (Kpsell, 2006), performance is important for users who are willing to use the system.

Wendolsky et al. (2007) have compared Tor with JAP. As conclusion, they mention that performance of Tor is similar to JAP, but they also mention that the performance is unpredictable but the bandwith and user tolerance for latency are better than in JAP. Furthermore, they conclude that this performance is good enough for Web surfing and downloads.

Compared with the other solutions for the TCP/IP layer (including I2P), this is the network that provide a better average bandwith (Abou-Tair et al., 2009; Schomburg, 2009).

Loesing et al. (2008) and Lenhard et al. (2009) studied the performance in hidden services. From these works we can point out that Lenhard et al. (2009) have found that the performance was worse than expected in low-bandwith networks (they identified mainly two problems: the download of relay descriptions in bootstrapping phase and the building or extension of virtual circuits when accessing to the hidden services) and they have proposed several solutions to improve it.

3.4.11. JAP/JonDo

Java Anon Proxy (JAP) - JonDo is the name of the commercial version - is an anonymous proxy that connects to a set of servers established as a cascade of Mixes (JonDonym).

JAP is the latest release (it is the client software) of the solution explained in Section 3.1.2 (Web MIXes) and developed in the AN.ON project (AN.ON Project, 2011) (JonDonym in the name used in the commercial version of the software). Although the project has the commercial version, we have decided to include it in this analysis because the software is free and some cascades can still be used freely.

This solution offers privacy at TCP/IP layer but it does not consider HTTP layer or application layer. Therefore, we should use it in combination with other tools. Some of these tools have also been developed in the AN.ON project, such as the JonDoFox browser.

The JonDonym's Web portal is also interesting because it contains an anoymity test (JonDonym, 2011a) that can inform a user on the different risks that her Web browser system is exposed when you are accessing to the Web. Furthermore, it provides you with some advice in order to solve your privacy flaws.

Currently, JonDo network is the smallest anonymity network (with 11 servers) and the least used (Li et al., 2011). In (Federrath, 2005) we can find some results about the use of this network (mainly for accessing to entertainment content such erotic, private homepages, games and services such as search engines, stock quotes, etc), regions of use (mainly Europe and Asia) and misuse (from law enforcement agencies and private complaints).

The usability of this tool has been analysed in (Abou-Tair et al., 2009) and they conclude that this tool is easy to use, although the distintion between JAP and JonDo should be clarified to avoid user's confusion. This tool has an interesting feature: it incorporates a visual anonymity meter that provides the user some information on her level of protection (Berthold et al., 2001; Clark et al., 2007). As for usability criteria (Abou-Tair et al., 2009), this tool has obtained better results than Tor and I2P.

With respect to performance, Wendolsky et al. (2007) mention that JAP with Jondonyms cascades is similar to Tor. They also conclude that the latency is less than in Tor and the quality of service that is perceived by users is more consistent than Tor. On the other hand, the throughout and user tolerance for latency are better in Tor than in JAP, but JAP is better than I2P (Abou-Tair et al., 2009).

3.4.12. JonDoFox

JonDoFox (JonDonym, 2011b) is a profile for Mozila Firefox that is optimized for secure anonymous surfing. It can be installed from the scratch (based on Mozilla Firefox portable) or on your own Firefox. This modified Web browser allows users to choose the proxy to be used (none, JAP/JonDo, Tor or a customized proxy) as well as it provides protection against of PII leakage.

JonDoFox can manage the following features related to the HTTP layer: referrer, user-agent, tools for cookies management (Cookie Monster - see Section 3.4.16). It also behaves as a Web browser with private browsing mode since it erases Web searches just after they are submitted, it also erases history periodically and offers protection against attacks to the cache in order to obtain cache cookies or Web pages previously visited.

At the application layer it allows the control of JavaScripts (initially they are disabled and you can indicate, for a particular provider, if you consider it as untrusted or if you grant them permanent or temporal permissions), Flash and plugins (in a similar way as JavaScript they are initially blocked and you can grant them some permissions). In this level it also supports the filter of advertisements.

JonDoFox combined with JAP/JonDo or Tor provides a comprehensive solution that covers all layers (TCP/IP, HTTP and application).

3.4.13. I2P

I2P (I2P, 2011) is the implementation of the solution presented in Section 3.1.3. This tool offers a way to se-

curely communicate network applications such as Web, mail, peer-to-peer, IRC chat, etc. It provides a graphical interface although it can also be used in command line mode, which is not recommended for novice users.

When the tool is installed, it does not offer any tool to users so that they can make their Web traffic to go through I2P network. The user has to configure manually the use of a proxy, an HTTP outproxy so that the IP address of the user is concealed.

The usability of this tool has been analysed in (Abou-Tair et al., 2009) and the authors conclude that this tool need to be improved in order to be used by novice users since its use requires technical knowledge for the installation and configuration processes.

As for usability criteria (Abou-Tair et al., 2009), this tool has obtained worse results than Tor and JAP/Jondo, which are more mature tools with a long way in this field.

Currently, this network is the second largest anonymity network (with 483 servers) after Tor (Li et al., 2011).

With respect to performance, (Abou-Tair et al., 2009) we can mention that I2P's average bandwith is worse than Tor and JAP.

As this tool only covers TCP/IP level, we should combine it with tools that offer protection in the other levels.

3.4.14. Firebug with Firecookie

Firebug (Hewitt et al., 2011a) is an add-on for Mozilla Firefox and Google Chrome that allows us to perform Web development tasks - for other Web browsers we can make use of Firebug Lite (Hewitt et al., 2011b) -.

We have included this tool since with it we can control all the information that is sent and received by our Web browser (HTTP request and responses, HTML, CSS, Javascript, etc). Furthermore, we can install an add-on for this tool named Firecookie in order to control cookies.

Firecookie (Odvarko, 2011) supports to inspect the cookies we receive, create them, remove them as well as to define permissions (if they are enabled or not, accept/deny cookies from a Web site, edit them, remove them, etc). Therefore, this tool only covers some limited protection at HTTP layer.

3.4.15. Cookies Manager+

Cookies Manager+ (V@no, 2011) is an add-on for the Mozilla Firefox, which allows a more advanced control of cookies than the Web browser provides. Namely, this tool allows us to view them (their values, when they were created and accessed) classified by domains, to edit and modify them, clear (all of) them, allow/block them, backup and restore them, even to add a cookie for a domain. Therefore, this tool only covers some limited protection at HTTP layer.

3.4.16. Cookie Monster

Cookie Monster (Schilling, 2011) is a Mozilla Firefox add-on that helps with the management of (session) cookies. It can show and manage first and third party cookies. The tool allows the acceptance, rejection and temporary acceptance of cookies. This management can be general for all Web sites or specified for specific sites. Therefore, this tool only offers privacy protection in a limited issue of HTTP layer.

3.4.17. CookieCuller

CookieCuller (Yamaoka, 2011) is a tool that facilitates the delete of non-desired cookies. With this tool we can establish the cookies to proctect and the rest of cookies can be deleted manually. We can also establish that once one cookie is deleted, this cannot be established again. Thus, this tool facilitates the management of specific cookies but it offers a quite reduce functionality as for the cookies management for privacy issues.

3.4.18. Adblock Plus and AdBlock

Adblock Plus (Palant, 2011) is an advertisement filter for Mozilla Firefox and Google Chrome that blocks all advertisements automatically. Thus, user navigation is faster and it can prevent some privacy issues related to tracking user by means of images of third party entities.

Adblock Plus allows the definition of filters with advanced features such as the use of regular expressions. This tool has received several awards (see (Palant, 2011)). Futhermore, we can download existing filter lists such as Easy List (Michael, Ares2, Erunno, Khrin and MonztA, 2011) or Fanboy list (Fanboy, 2011) in order to facilitate the user the definition of filter lists that automatically avoid advertisements (even those that are placed in videos), banners and tracking. These lists can also be used with Microsoft Internet Explorer. Thus, Adblock Plus is designed to protect the user from the risks of the application layer as for active content as advertisements. It should be combined with other tools for this level as well as some tools for the other levels.

As Adblock Plus was not available for Google Chrome, AdBlock was created (Weisbein, 2011). AdBlock is available for Safari (Gundlach, 2011b) and Google Chrome (Gundlach, 2011a). AdBlock is inspired in Adblock Plus (ABP) and shares most of the features that ABP offers: it blocks advertisements and it allows the use and definition of different kind of filters.

Unlike ABP, AdBlock offers some new additional features such blocking advertisements in Flash games or hiding a section in a Web page.

In order to know how good your popup blocker is you can use the different tests that are provided by *PopupTest.com* (WebAttack, Inc, 2011).

3.4.19. ChromeBlock

ChromeBlock (Abine, 2011a) is an extension for Google Chrome that blocks Web beacons, bugs, advertisers and establishes opt-out cookies. Furthermore, it is important to point that it supports that user can be informed and she can manage how it is tracked in each Web site. Thus, users can decide on how much information they provide on their Web behaviour. Therefore, this solution provides privacy at HTTP and applications layers.

3.4.20. PithHelmet

PithHelmet (Solomon, 2011) is an ad blocker plug-in for Safari. Hence, it can block ad images and Flash. With this tool is possible to configure both the ad blocking level and the cookie privacy level. Furthermore, it supports the specification of rules (based on Perl-Compatible Regular Expressions) that define the content to be blocked as well as blocking images/cookies from specific Web sites. Therefore, this solution is used to provide privacy at HTTP and applications layers.

3.4.21. NoScript

NoScript (Maone, 2011) is a tool for blocking active content such as Java, JavaScript, Flash, Silverlight, Web bugs, plugins, etc.

The active content is blocked by default. However, the tool offers the possibility of defining trusted Web sites as well as you can decide that the scripts of a Web site can be executed temporarily or permanently. It also support the Do Not Track (see Section 3.2.5). Thus, it covers some solutions of HTTP layer (Do Not Track) and application layer (being a HTML filter).

3.4.22. JavaScript blacklist

JavaScript blacklist (Thaler, 2011) is an extension which can block Javascript from a list of domains that can be configured. Thus, this tool only offers protection at application level.

3.4.23. Ghostery

Ghostery (Cancel and Shnir, 2011) is a tool that detects and blocks Web bugs, scripts and trackers (ad networks, behavioral data providers, Web publishers, etc). It also allows the user to know the information each Web site gathers and the privacy policy that follows.

An interesting option of Ghostery is the possiblity of informing on the companies that track information on users. This information will be stored in a server that can be used for the tool in order to improve progressively the control of the trackers. Thus, this tool helps in improving privacy at application level.

3.4.24. BetterPrivacy

BetterPrivacy (Yardley, 2011) protects against Flash cookies (Local Shared Object - LSO). This add-on for Mozilla Firefox erases Flash cookies on the exit of the Web browser. It also allows the visualization and management of this kind of cookies. We can also protect those that we are interesting in. Futhermore, it offers different options for configuring the erase of these cookies: time, on exit, on application start, etc. Therefore, this solution offers some protection as for application layer.

3.4.25. OptimizeGoogle

OptimizeGoogle (OptimizeGoogle, 2011) is an add-on for Mozilla Firefox which purpose is to optimize the results that Google returns when a query is made. Furthermore, it removes annoying content and protect user's privacy. Namely, this tool blocks Google Analytics cookies, removes advertisements and click tracking and anonymizes user's Google identifier. Therefore, this tool contributes in the protection of user's privacy at application level but it is limited to Google and it would be useful to be used with other search engines as Yahoo, Bing, etc.

3.4.26. TrackMeNot

TrackMeNot (Howe and Nissenbaum, 2009; Howe et al., 2011) is a Web browser add-on that aims to prevent that Web search engines can create a user profile from the Web searches the user makes. To achieve this goal this tool uses obfuscation techniques by issuing periodically search queries generated randomly. Hence, the real user queries are mixed in a crowd of other queries, which makes more difficult the creation of user profiles. This queries can be sent through different Web search engines such as Google, Bing, AOL, Yahoo!, etc. Therefore, this solution contributes to protect user privacy at application layer for Web search engines.

3.4.27. Starting page

Starting page (Abine, 2011b) is a private Web search engine that uses Google to make queries at the same time they protect your privacy.

This engine does not create a log with your IP address and it does not either use tracking cookies. These facts have been certified with the European Privacy seal (EuroPriSe, 2011) and by Certified Secure.

This Web search engine also offers the possibility of using SSL/TLS for the HTTP connection. Furthermore, it also provides plug-ins for the main Web browsers. Therefore, this Web search engine allows users to protect privacy at application level for Web search queries.

3.4.28. Scroogle

Scroogle (Scroogle, 2011) is a private Web search engine based on Google that avoids that Google can track user by means of cookies or user's IP address. Furthermore, logs are deleted after 48 hours.

User Web search queries can be sent to Scroogle by using SSL/TLS. When this search engine receives user's query, it chooses randomly an IP address between seven hundred possible addresses and this chosen address is used to send the request to Google. The cookie that Google establishes when the results to the query are provided is deleted. These results are provided to the user and Scroogle deletes them after an hour. Thus, this Web search engine allows user to protect privacy at application level for Web search queries.

4. Comparison and discussion

In this section we perform a comparison between the different solutions that the tools analysed previously provide.

For each tool we compare different features such as: OS supported, type of license, latest release (month and year) and which layers the tool covers as for privacy.

We also compare the different privacy protection features that each browser offers as well as the different extension/plug-ins/add-ons that can be incorporated to them in order to improve these features.

This information is shown by means of different tables. Hence, firstly, we compare the different tools from its current state, that is, which OS support, the type of license under the software is distributed and, finaly, when the latest version of the tool has been released. Thus, users can know whether they can use it in their operating system, the conditions of the license and whether the tool is updated. This information in shown in Table 1.

As we can see in that table, we have included the main Web browsers -see report from StatCounter (StatCounter, 2011) and Chikita Insights (Cavanagh, 2011)-. We have included them since although they are not a privacy tool, they are used to browse Web pages and they contain some elements (configuration options) that help in providing a better privacy when users are surfing on the Internet.

In Table 1 we can also see that except for Multiproxy that is only available for Windows, the rest of tools are available for the main current OS (Linux, Windows and Mac OS). This is also due to the fact that most of the tools are extensions to Web browsers, thus, if the Web browser is developed for an OS, in general, the extension automatically works for that OS.

From Table 1 we also want to mention that although Microsoft Internet Explorer has a propietary license, we have included it since if the user has a Windows license, then, its download is free.

We can also see that most of the tools covered have released recently (in the last six months) a version of its software.

Next, in Tables 2, 3 and 4 we show the different configuration options that main Web browsers offer to end users in order to improve their privacy protection.

In Table 2 we compare the different options they offer with respect to allow/block different Web objects (pop-ups, JavaScript, Java, ActiveX and Web bug).

As we can see in Table 2, all browsers support the blocking of pop-ups. It is important to point out that only Microsoft Internet Explorer considers Web bugs, although with the blocking of images this problem could be solved in other Web browsers. However, this can cause usability problems to users.

We can also mention that all of them support the blocking of Javascript. However, not all the Web browsers can block Javascript for a specific Web site.

In Table 3 we compare how the different Web browsers support the management of cookies (both first and third party cookies) as well as Do Not Track feature.

Related to cookies, in general, current Web browsers offer a wide range of options and in most of the cases you can perform its management in an individual level for each Web site (see Table 3). This is due to the fact that cookies management is a fundamental element to maintain privacy.

Both Mozilla Firefox and Microsoft Internet Explorer support for prompting for cookies. This option can be useful for advanced users that can decide whether to accept a cookie or not. However, this option, if enabled, can be disturbing since, in general, the most popular Web sites use more than a cookie (Yue et al., 2007, 2010).

The Do Not Track feature is supported by all Web browsers except for Google Chrome as they do not consider this approach suitable and, therefore, they have decided not support it. Instead of it, their approach is to

Tool	OS			License	Latest Release
	Linux	Windows	Mac OS		
Microsoft Internet Explorer 9	Y	Y	Y	Propietary	8/2011
Mozilla Firefox 7	Y	Y	Y	MPL/GPL/LGPL	9/2011
Google Chrome 12	Y	Y	Y	BSD, Google Chrome Terms of service	9/2011
Safari 5	Y	Y	Y	Propietary, LGPL	7/2011
Multiproxy	N	Y	N	Free	12/2008
CGIproxy	Y	Y	Y	Free	12/2008
Privoxy	Y	Y	Y	GNU GPLv2	11/2010
Polipo	Y	Y	Y	MIT	5/2011
Tor	Y	Y	Y	BSD	2/2011
Torbutton	Y	Y	Y	BSD	7/2011
FoxyProxy	Y	Y	Y	GNU GPLv2	7/2011
UnPlug	Y	Y	Y	Affero GPL license v3	8/2011
Plugin customs	Y	Y	Y	Free	10/2010
Tor browser bundle	Y	Y	Y	BSD	7/2011
JAP/JonDo	Y	Y	Y	BSD	7/2011
JonDoFox	Y	Y	Y	BSD	7/2011
I2P	Y	Y	Y	BSD	6/2011
Firecookie	Y	Y	Y	BSD	8/2011
Cookies Manager+	Y	Y	Y	Mozilla Public License, version 1.1	4/2010
Cookie Monster	Y	Y	Y	Mozilla Public License v1.1	10/2010
CookieCuller	Y	Y	Y	Mozilla Public License v1.1	10/2010
Adblock Plus	Y	Y	Y	Mozilla Public License v1.1	6/2011
Adblock for Safari	N	Y	Y	GNU GPL v3	6/2011
Adblock for Chrome	Y	Y	Y	GNU GPL v3	6/2011
ChromeBlock	Y	Y	Y	Free	7/2011
PithHelmet	Y	Y	Y	GNU GPL v2	6/2011
NoScript	Y	Y	Y	GNU GPLv2	8/2011
JavaScript blacklist	Y	Y	Y	Free	N/A[a]
Ghostery	Y	Y	Y	GNU GPLv2	9/2011
BetterPrivacy	Y	Y	Y	GNU GPLv2	8/2011
OptimizeGoogle	Y	Y	Y	GNU GPL	11/2010
TrackMeNot	Y	Y	Y	Creative Commons	7/2011
Starting page	Y	Y	Y	Free	-[b]
Scroogle	Y	Y	Y	Free	-[b]

[a] Information not available
[b] It is a Web site

Table 1: Tools, OS, license and latest release

Option \ Web browser	Mozilla Firefox	Microsoft Internet Explorer	Google Chrome	Safari
Allow/Block pop-up windows	x	x	x	x
Allow/Block pop-up windows for specific Web sites	x	x	x	x
Allow/Block load images automatically	x	x	x	
Allow/Block load images from particular Web sites	x	x	x	
Allow/Block Web bugs		x		
Allow/Block Web bugs for particular Web sites		x		
Allow/Block Javascript	x	x	x	x
Allow/Block Javascript for specific Web sites		x	x	
Allow/Block Javascript move or resize existing windows	x			
Block/Allow Javascript raise or lower windows	x			
Allow/Block Javascript disable or replace context windows	x			
Allow/Block ActiveX	*[a]	x		
Allow/Block Java	*[b]	*[b]		x
Enable/disable extensions/plug-ins/add-ons	x	x	x	x
Enable/disable an specific extension/plug-in/add-on for private browsing mode			x	
Automatically block tracking content (scripts, images, ads)		x		

[a] Mozilla Firefox does not support ActiveX

[b] By default it does not support Java and it has to be included as an extension. Thus, it is disabled as an extension.

Table 2: Comparison of Web browsers as for Web objects management

Option \ Web browser	Mozilla Firefox	Microsoft Internet Explorer	Google Chrome	Safari
Accept/block first-party cookies	x	x	x	x
Accept/block first-party cookies from particular Web sites	x	x	x	
Accept/block third-party cookies	x	x		x
Accept/block third-party cookies from particular Web sites	x	x	x	
Prompt first-party cookies	x	x		
Prompt third-party cookies	x	x		
Always allows session cookies		x		
Delete all cookies stored	x	x	x	x
Delete stored specific cookies	x		x	x
Do Not Track	x	x		

Table 3: Comparison of Web browsers as for cookie management and Do Not Track

support opt-out cookies.

In Table 4 we present the results of the comparison between the different Web browsers as for management of history, passwords, private browsing mode, geolocation and the choice of the language to show Web pages. These are additional elements that can reveal user's PII to both a local attacker and Web attacker Aggarwal et al. (2010).

All the Web browsers support private browsing mode since they are conscious that privacy is an important feature for users. However, it is important to point out that there are differences between the diverse ways of implementing private browsing mode as mentioned in (Aggarwal et al., 2010). Apart from this feature that, in general, controls cookies, history and plug-ins so that they do not share information between different sessions in the browser and they delete PII once the session has finished. Private browsing mode in these Web browsers also allows the customization of how to manage the elements previously mentioned as we have seen in the previous tables. Furthermore, the management of these options can be improved with the tools mentioned throughout Section 3.2 and that are shown in Tables 5 and 6.

In Table 4 it is important to point out the capability of users to configure the Web browser to block those Web sites that are being notified as Web site forgeries or reported attack sites. This is specially useful for end users since they can know dangerous sites without being experts and help them in to decide which Web sites to trust in.

It is also worth to mention geolocation option since nowadays many Web sites use the geolocation API (Application Programming Interface) for obtaining user's location (this can be used to create user's profiles). However, with the possibility of enabling or disabling this option users can be aware of the Web sites that request their location and they can decide whether they provide their location.

Finally, in Table 4 we can see that, except Safari, all of them allow the choice of the language. Thus, if we configure that our chosen language is English we will difficult locate the user, fingerprinting our Web browser and, therefore, the creation of user's profiles.

As we can see in Tables 2, 3 and 4, Web browsers also allow the user can control how to manage cookies, history, passwords and establish exceptions for each of these elements both in private browsing mode and in public mode. The capability of being able to define exceptions allows users to establish a better control of the Web sites they want filter and mantain private at the same time they maintain usability in navigation.

These tables also show that the Web browsers that allow a better control on the different elements that could cause a leakage of PII are Mozilla Firefox and Microsoft Internet Explorer, followed by Google Chrome and Safari.

In Table 5 we show the different tools and how these are available for the different Web browsers. In this table we can see that most of free tools are available for Mozilla Firefox. We can also point out that only a few of them are available for all the Web browsers analysed: Firecookie, Ghostery and private browsing mode. Apart from these tools, we can mention Starting page and Scroogle because they are Web sites. We have not included Tor, JAP/Jondo, I2P, Multiproxy, CGIproxy, Privoxy and Polipo because they are tools that are independent of the Web browser to use.

From the different tools analysed and freely available we can mention that the largest group is devoted to cookies management, followed by those that are for proxies, pop-ups and Web bugs. On the other hand, the features that are less covered are: Do Not Track and private Web search.

As for private Web search we can also mention that, from the tools analysed, all support the use of Google as Web search engine. However, only Trackmenot allows the use of other Web search engines different from Google.

We can also mention that most of the tools are centered on protecting an specific level (TCP/IP, HTTP or application). The tools that cover more than one level are: Privoxy, Polipo, TorButton, Tor browser bundle, JonDoFox, PithHelmet and Optimize Google.

In particular, the tools that cover more levels are Tor browser bundle, Privoxy and Polipo. Indeed, they cover all of them.

In the case of Tor browser bundle, this is due to the fact that is a combination of several tools. Even though, it does not cover all the elements to protect. Namely, this package does not include a private Web search tool. This could be solved by means of the installation of some of the extensions mentioned for this purpose (TrackMeNot, OptimizeGoogle or Starting page) or by using some of the Web sites mentioned for this purpose (Starting page or Scroogle).

In the case of Privoxy and Polipo, as for TCP/IP level, they do not protect the traffic eavesdropping, they only hide IP address if they are executed in another computer different from user's computer. These tools are proposed to be used in combination with Tor in order to avoid problems related to the other levels (mainly in the application layer as HTTP or HTML filter).

We can also point out the case of JondoFox that is

Option \ Web browser	Mozilla Firefox	Microsoft Internet Explorer	Google Chrome	Safari
Remember history	x			
Never Remember history	x			
Delete history	x	x	x	x
Customized remember of history	x			
Remember/Not remember browsing history	x			
Delete browsing history when browser closes	x	x		
Remember/Not remember download history	x			
Remember/Not remember search and form history	x			
Private Browsing mode	x	x	x	x
Block reported attack sites	x	x	x	
Block reported Web forgeries	x	x	x	
Allow/Block remember passwords	x			
Allow/Block remember passwords for particular Web sites	x			
Management of saved passwords	x		x	
Enable/disable geolocation	x	x		x
Enable/disable geolocation for specific Web sites			x	
Prompt for geolocation			x	
Choice of the language to read Web pages	x	x	x	

Table 4: Comparison of Web browsers as for management of history, passwords, private browsing mode, geolocation and language

a tool, which covers almost all the features required for navigate privately on the Web except those realted to the TCP/IP level. JondoFox as Tor browser bundle covers almost of them because it is a combination of several tools. In order to support all the features, we can combine it with JAP/Jondo or with Tor. Thus, the TCP/IP level would be covered with protection against traffic analysis.

Next, we discuss, for each Web browser, the different tools that we would need to cover all (or many of them as possible) the features analysed. In this analysis we also try to choose the combination of tools that requires the least number of tools within the set with the aim of facilitating the installation process to end users.

In all combinations the use of Stating page is proposed. As for cookies management and pop-ups, the tools to choose change. We can also mention that the management of Flash cookies is hardly supported and more tools are required.

For Mozilla Firefox the main combinations are: JondoFox and BetterPrivacy combined with JAP/JonDo or Tor or I2P, or Tor browser bundle and BetterPrivacy with Starting page/TrackMeNot plug-in.

The former combination consists of JondoFox that covers HTTP and application levels. Therefore, it only only needs to cover TCP/IP level. For this purpose we can use JAP/JonDo, Tor or I2P. Furthermore, it includes BetterPrivacy in order to remove Flash cookies.

The latter combination consists of Tor browser bundle that covers the three levels. However, in the third level it does not include a plug-in for private Web search. For this purpose, the use of Starting page/TrackMeNot plug-in is recommended. Although we could also configure the Web browser so that Web search requests go through Scroogle's CGI. Finally, in the same way as the previous combination, it also includes BetterPrivacy for the control of Flash cookies. Thus, with these tools we can cover all the features mentioned.

For Microsoft Internet Explorer the set of tools that could cover most of the features of the different levels are: Tor or JAP/Jondo or I2P, Polipo, AdBlockIE, FoxyProxy, Ghostery and Starting page. These tools cover most of the privacy protection features of the three level. However, in this combination there is no tool that can remove automatically Flash cookies.

For Google Chrome the set of tools that could cover most of the features of the different levels are: Tor or JAP/Jondo or I2P, Polipo, FoxyProxy, ChromeBlock, BetterPrivacy and Starting page. In this set of tools the support of Do Not Track is not provided.

In Safari the set of tools chosen is: Tor or JAP/Jondo

Tool	Web browser			
	Mozilla Firefox	Microsoft Internet Explorer	Google Chrome	Safari
TorButton	x			
FoxyProxy	x	x[a]	x[a]	
UnPlug	x			
Plugin customs				x
Firecookie	x	x	x	x
Cookies Manager+	x			
Cookie Monster	x			
CookieCuller	x			
Adblock Plus	x	x		
Adblock			x	x
AdblockIE		x		
ChromeBlock			x	
PithHelmet				x
NoScript	x			
JavaScript Blacklist				x
Ghostery	x	x	x	x
BetterPrivacy	x		x	
Private Browsing Mode	x	x	x	x
OptimizeGoogle	x			
TrackMeNot	x		x	
Starting page	x[b]	x	x[b]	x
Scroogle	x[c]	x[c]	x[c]	x

[a] Available soon.

[b] It is a Web site, therefore it works with any Web browser. Furthermore, for this Web browser it also offers a plug-in.

[c] It is a Web site, therefore it works with any Web browser. Furthermore, for this Web browser it also indicates how to customize the browser so that search queries go directly to this Web search engine.

Table 5: Tools for Web browsers

or I2P, Polipo, Unplug, Plugin customs/PithHelmet, JavaScript Blacklist and Starting page. Although most of the features are coverded for the three levels, in this combination there is no tool that can remove automatically Flash cookies.

From these combinations we can point out that the tools for mixes in TCP/IP level can be used with any Web browser since they are independent of the browser and are used through a proxy. We can also point out Polipo as HTTP/HTML filter.

Finally, from out analysis, we can derive that currently for the main Web browsers there are enough tools to navigate privately on the Web, although some issues related to usability and performance should be improved.

5. Related Work

There is an important number of proposals that have been designed to perform anonymous communications. In general, the different works and surveys that have analysed the state of the art of in this field do not have analysed all the elements that are needed to perform an anonymous Web communication (see the different levels to cover in Section 2.2). In general, these works have focused on the analysis of the solutions for some on the levels we introduced. Next, we mention those works and the different solutions they analyse.

For the analysis of anonymous communications at TCP/IP level, which is the topic most analysed, there are several works that are a reference in the field (Rezgui et al., 2003; Gritzalis, 2004; Danezis and Diaz, 2008; Danezis et al., 2009; Edman and Yener, 2009; Danezis and Gürses, 2010; Ren and Wu, 2010).

The use of cookies and different solutions has been mentioned in (Rezgui et al., 2003; Senicar et al., 2003; Linn, 2005; Yue et al., 2010; Barth, 2011). Although an exhaustive comparison between different proposal has not been performed.

An analysis of private browsing mode in the different Web browsers has been presented in (Aggarwal et al., 2010). Namely, this paper compares how this mode is supported in Mozilla Firefox, Microsoft Internet Explorer, Google Chrome and Safari.

We can also find some works related to the usability of anonymity networks and tools in (Clark et al., 2007; Abou-Tair et al., 2009; Schomburg, 2009; Fabian et al., 2010). In these papers we can find data on the number of users, countries and the main difficulties users have when they want to navigate privately with some of the tools available.

In spite of these works, to the best of our knowledge, there is no work that considers all the different protection measures that should be taken into account when a user is surfing on the Web. There are works that only cover a part of the whole problem as we have just mentioned. Thus, in order to offer a comprehensive view, in this paper we have analysed which anonymous communication methods have been implemented, which tools support them and how they can be combined in order to be used to surf privately on the Web.

In our paper we have also shown the relationships between the different risks in the different layers, e.g., if we provide privacy at TCP/IP level, we have seen that user's PII could be compromised by menas of cookies or Web browser fingerprinting, that is, based on the information of the other levels. Therefore, we need to provide protection in the three levels.

Apart from the risks, we have also shown how the different techniques and tools can be combined for minimize the risks users are exposed when they are surfing on the Web. This have been studied and analysed for the main Web browsers, that is, Mozilla Firefox, Microsoft Internet Explorer, Google Chrome and Safari (from Apple).

6. Conclusions and future work

Users are concerned for privacy when they surf on the Internet. Indded, increasingly Web users are realising of the importance that companies know information on their preferences, behaviour, purchase habits, etc. As a consequence, in the last ten years we have seen how scientific community has researched in this field to offer this kind of solutions to this problem and an important number of proposals have appeared in order to provide anonymous communications on the Internet.

In order to navigate privately on the Web it is required that the development of solutions that take into account different levels where personally identificable information leakage could happen. Namely, privacy can be compromised using information of three different levels: TCP/IP level, HTTP level and application level.

In this paper we have described the different risks associated to each level, the different techniques that have been proposed and, from those that have been developed, we have analysed them in order to know the different advantages, disadvantages and possible attacks could happen. This analysis shows that privacy is a complex issue and that we need to combine different techniques for each level in order to provide a comprehensive solution that do not compromise user's privacy.

Tool	Level												
	TCP/IP		HTTP					Application					
	Proxy	Mixes	Cookies	HTTP filter	SAWP[a]	PBM[b]	DNT[c]	Pop ups	Web bugs	Scripting	Active Objects[d]	AAWP[e]	PWST[f]
Multiproxy	x												
CGIproxy	x												
Privoxy	x[f]		x	x	x			x	x	x	x	x	
Polipo	x[f]		x	x	x				x	x	x	x	
Tor		x											
TorButton			x	x		x		x	x	x	x		
FoxyProxy	x												
UnPlug											x[g]		
Plugin customs											x		
Vidalia bundle	x	x											
Tor browser bundle	x	x	x	x	x	x	x	x	x	x	x	x	
JAP/JonDo		x											
JonDoFox			x	x	x	x	x	x	x	x	x	x	x
I2P		x											
Firecookie			x										
Cookies Manager+			x										
Cookie Monster			x										
Cookie Culler			x										
Adblock Plus								x	x		x		
Adblock[g]								x	x		x		
ChromeBlock									x				
PithHelmet			x					x	x		x[h]		
NoScript								x	x	x	x		
JavaScript blacklist											x		
Ghostery									x	x			
Better Privacy											x[i]		
Optimize Google			x				x	x					x
TrackMeNot													x
Starting page	x[j]												x
Scroogle	x[j]												x

[a] Simple Anonymous Web Proxy
[b] Private Browsing Mode
[c] Do Not Track
[d] Flash, Activex, Java, etc
[e] Advanced Anonymous Web Proxy
[e] Private Web Search Tool
[f] It hides IP if it executed in a host different from user's computer
[g] Both AdBlock for Google Chrome and AdBlock for Safari
[h] Only for Flash
[i] Only for Flash cookies
[j] From the point of view of the Web search engine they are a proxy.

Table 6: Privacy levels covered

These techniques have been developed by different software tools. We have made an analysis on the main tools that are freely available, that, as we have shown, there are an interesting number of them.

From the analysis made on these tools, we can point out that there is no single software package that facilitates users navigate privately. Even though, there are some packages such as Tor browser bundle and JondoFox that cover many on them and only need some additional tools to provide a comprehensive solution that protects user's PII.

In this paper we have identified the main combinations of tools available for each Web browser in order to provide a comprehensive solution. This is an issue that should be taken into account in order to facilitate that users can surf privately on the Web and increase their usability. Thus, a suite or package that installed easily the different combination of tools proposed, it would facilitate its acceptance. Currently, end user would have to install, at least, three software tools (it depends on the Web browser chosen).

This paper also shows that Mozilla Firefox is the Web browser that has more tools and options to configure any of the different features to be covered in each level. In spite of the fact, for the other Web browsers also exist tools that could cover almost all the features required to surf on the Web privately. However, they require the installation of different tools can suppose usability problems for end users as we have just mention. Mainly, this is due to the fact that this installation and configuration is sometimes difficult or not understable for end users.

Therefore, this paper, thanks to the analysis made on the different techniques, tools and the levels to cover in order to protect users' privacy, provides a comprehensive view to both researchers and end users on the privacy risks when surf on the Web and how they can mitigated thanks to the use of different tools that are available for the main Web browsers.

7. Acknowledgements

This work has been partially funded by the "Seneca Foundation for Excellent Group in the Region 04552/GERM/06", TIN2008-06441-C02-02 "Infraestructure of ubquitous services and communications" and CIP-ICT PSP-2009-3 "Secure management of information across multiple stakeholders" projects.

References

Abine . Chitika Insights. Web Browser Market Share: September, 2011 Update. Web site: `https://chrome.google.com/webstore/detail/epanfjkfahimkgomnigadpkobaefekcd` [Last accessed: 01 October 2011]; 2011a.

Abine . Chitika Insights. Web Browser Market Share: September, 2011 Update. Web site: `https://chrome.google.com/webstore/detail/epanfjkfahimkgomnigadpkobaefekcd` [Last accessed: 01 October 2011]; 2011b.

Abou-Tair De, Pimenidis L, Schomburg J, Westermann B. Usability Inspection of Anonymity Networks; IEEE. p. 100–9.

Aggarwal G, Bursztein E, Jackson C, Boneh D. An analysis of private browsing modes in modern browsers; USENIX Association. USENIX Security'10. p. 1–6.

AllNetTools . Smart Whois. Web site: `http://www.all-nettools.com/toolbox/smart-whois.php` [Last accessed: 15 August 2011]; 2011.

Alsaid A, Martin D. Detecting web bugs with bugnosis: privacy advocacy through education; Springer-Verlag. PET'02. p. 13–26.

AN.ON Project . Project: AN.ON-Anonymity.Online. Web site: `http://anon.inf.tu-dresden.de/index\_en.html` [Last accessed: 15 August 2011]; 2011.

Ardagna CA, Camenisch J, Kohlweiss M, Leenes R, Neven G, Priem B, Samarati P, Sommer D, Verdicchio M. Exploiting cryptography for privacy-enhanced access control: A result of the prime project. J Comput Secur 2010;18(1):123–60.

Back A, Möller U, Stiglic A. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems; Springer-Verlag. IHW '01. p. 245–57.

Barth A. HTTP State Management Mechanism. RFC 6265; RFC Editor; 2011.

Baviskar S, Thilagam PS. Protection of web user's privacy by securing browser from web privacy attacks. International Journal of Computer Technology and Applications 2011;2(4):1051–7.

Behl S, Lilien L. Privacy-preserving transactions on the web. The Hilltop Review 2009;3(1):14–38.

Berthold O, Federrath H, Köhntopp M. Project "Anonymity and unobservability in the Internet"; ACM. CFP'00. p. 57–65. ACM ID: 332211.

Berthold O, Federrath H, Köpsell S. Web MIXes: A System for Anonymous and Unobservable Internet Access; Springer Berlin Heidelberg; volume 2009. p. 115–29.

Bissias GD, Liberatore M, Jensen D, Levine BN. Privacy vulnerabilities in encrypted http streams. In Proceedings of Privacy Enhancing Technologies Workshop (PET 2005 2005;:1–11.

Cancel D, Shnir F. Ghostery. Web site: `http://www.ghostery.com` [Last accessed: 15 August 2011]; 2011.

Carroll TE, Grosu D. A secure and anonymous voter-controlled election scheme. Journal of Network and Computer Applications 2009;32(3):599–606.

Casado M, Freedman MJ. Peering through the shroud: the effect of edge opacity on ip-based client identification; USENIX Association. NSDI07. p. 13–27.

Castella-Roca J, Viejo A, Herrera-Joancomarti J. Preserving user's privacy in web search engines. Computer Communications 2009;32(13-14):1541–51.

Cavanagh R. Chitika Insights. Web Browser Market Share: September, 2011 Update. Web site: `http://insights.chitika.com/2011/web-browser-market-share-september-2011-update/` [Last accessed: 01 October 2011]; 2011.

Chaabane A, Manils P, Kaafar MA. Digging into Anonymous Traffic: A Deep Analysis of the Tor Anonymizing Network; IEEE Computer Society. NSS '10. p. 167–74.

Chairunnanda P, Pham N, Hengartner U. Privacy: Gone with the Typing! Identifying Web Users by Their Typing Pattern; Springer. 4th Hot Topics in Privacy Enhancing Technologies (HotPETs 2011). p. 1–15.

Chaum DL. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 1981;24:84–90. ACM ID: 358563.

Chen T, Fu J. Protection of privacy on the Web; IGI Global.

Chroboczek J. Polipo a caching web proxy. Web site: http://www.pps.jussieu.fr/~jch/software/polipo/ [Last accessed: 15 August 2011]; 2010.

Chudnov A, Naumann DA. Information flow monitor inlining. In: Proceedings of the 2010 23rd IEEE Computer Security Foundations Symposium. Washington, DC, USA: IEEE Computer Society; CSF '10; 2010. p. 200–14.

CISCO Systems I. Online privacy - How to protect yourself and your family, 2009.

Clark J, van Oorschot PC, Adams C. Usability of anonymous web browsing: an examination of Tor interfaces and deployability; ACM. SOUPS 07. p. 41–51.

Cookies.org AA. All About Cookies. Web site: http://www.allaboutcookies.org [Last accessed: 15 August 2011]; 2011.

Danezis G, Clayton R. Introducing Traffic Analysis; Auerbach Publications. p. 95–117.

Danezis G, Diaz C. A Survey of Anonymous Communication Channels. Number MSR-TR-2008-35, 2008.

Danezis G, Diaz C, Syverson P. Systems for anonymous communication; Chapman & Hall/CRC. CRC Cryptography and Network Security Series. p. 341–89.

Danezis G, Gürses S. A critical review of 10 years of Privacy Technology. p. 1–16.

Dbatley . UnPlug. Web site: http://unplug.dbatley.com [Last accessed: 15 August 2011]; 2011.

Dhawan M, Ganapathy V. Analyzing information flow in javascript-based browser extensions. Computer Security Applications Conference, Annual 2009;0:382–91.

Dingledine R, Mathewson N, Syverson P. Tor: the second-generation onion router; USENIX Association. SSYM'04. p. 21–. ACM ID: 1251396.

Dolipo . Dolipo GUI wrapper for osx. Web site: http://code.google.com/p/dolipo/ [Last accessed: 15 August 2011]; 2008.

Eckersley P. How unique is your web browser? In: Atallah MJ, Hopper NJ, editors. Privacy Enhancing Technologies. Springer; volume 6205 of *Lecture Notes in Computer Science*; 2010. p. 1–18.

Edman M, Yener B. On anonymity in an electronic society: A survey of anonymous communication systems. ACM Computing Surveys 2009;42(1):1–35.

Electronic Frontier Foundation . Panopticlick. Web site: https://panopticlick.eff.org/ [Last accessed: 15 August 2011]; 2011.

EuroPriSe . EuroPriSe - the European Privacy Seal for IT Products and IT-Based Services. Web site: https://www.european-privacy-seal.eu [Last accessed: 15 August 2011]; 2011.

Fabian B, Goertz F, Kunz S, Mller S, Nitzsche M. Privately Waiting - A Usability Analysis of the Tor Anonymity Network; Springer Berlin Heidelberg; volume 58. p. 63–75.

Fanboy . Fanboy List. Web site: http://fanboy.co.nz/ [Last accessed: 15 August 2011]; 2011.

Federrath H. Privacy Enhanced Technologies: Methods Markets Misuse; Springer Berlin Heidelberg; volume 3592 of *Lecture Notes in Computer Science*. p. 1–9.

Fielding R, Irvine U, Gettys J, Mogul J, Frystyk H, Masinter L, Leach P, Berners-Lee T. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616; RFC Editor; 1999.

Fonseca F, Pinto R, Meira Jr. W. Increasing User's Privacy Control through Flexible Web Bug Detection; IEEE Computer Society. p. 205–12.

Gabber E, Gibbons PB, Kristol DM, Matias Y, Mayer A. Consistent, yet anonymous, web access with lpwa. Commun ACM 1999;42(2):42–7. ACM ID: 293447.

Gemal H. BrowserSPY. Web site: http://browserspy.dk [Last accessed: 15 August 2011]; 2011.

Goldschlag DM, Reed MG, Syverson PF. Hiding Routing Information; Springer-Verlag. p. 137–50.

Golembiewski C, Hansen M, Steinbrecher S. Experiences Running a Web Anonymising Service; IEEE Computer Society. DEXA 03. p. 482–6.

Gritzalis S. Enhancing web privacy and anonymity in the digital era. Inf Manag Comput Security 2004;12(3):255–87.

Gross JB, Rosson MB. End user concern about security and privacy threats; ACM. SOUPS 07. p. 167–8.

Gulyas G, Schulcz R, Imre S. Comprehensive Analysis of Web Privacy and Anonymous Web Browsers: Are Next Generation Services Based on Collaborative Filtering? p. 17–32.

Gundlach M. AdBlock for Chrome. Web site: http://chromeadblock.com [Last accessed: 15 August 2011]; 2011a.

Gundlach M. AdBlock for Safari. Web site: http://safariadblock.com [Last accessed: 15 August 2011]; 2011b.

Harding W, Reed A, Gray R. Cookies and Web Bugs. What They Are and How They Work Together; CRC Press. p. 2133–42.

Herrmann D, Wendolsky R, Federrath H. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier; ACM. CCSW '09. p. 31–42.

Herrmann M, Grothoff C. Privacy-implications of performance-based peer selection by onion-routers: A real-world case study using i2p. In: Privacy Enhancing Technologies Symposium (PETS 2011). Springer Verlag; Waterloo, Canada: Springer Verlag; 2011. .

Hewitt J, Odvarko J, johnjbarton , robcee , FirebugWorkingGroup . Firebug. Web Development Evolved. Web site: http://getfirebug.com [Last accessed: 15 August 2011]; 2011a.

Hewitt J, Odvarko J, johnjbarton , robcee , FirebugWorkingGroup . Firebug. Web Development Evolved. Web site: https://getfirebug.com/firebuglite [Last accessed: 15 August 2011]; 2011b.

Hintz A. Fingerprinting websites using traffic analysis. IN WORKSHOP ON PRIVACY ENHANCING TECHNOLOGIES 2002;:171–8.

Hopper N, Vasserman EY, Chan-Tin E. How much anonymity does network latency leak?; ACM. CCS 07. p. 82–91.

Hopper N, Vasserman EY, Chan-TIN E. How much anonymity does network latency leak? ACM Trans Inf Syst Secur 2010;13(2):13:1–13:28.

Howe D, Nissenbaum H. TrackMeNot: Resisting Surveillance in Web Search; Oxford University Press. p. 417–36.

Howe DC, Nissenbaum H, Toubiana V. TrackMeNot. Web site: http://cs.nyu.edu/trackmenot/ [Last accessed: 15 August 2011]; 2011.

I2P . I2P Documentation. Web site: http://www.i2p2.de/how.html [Last accessed: 15 August 2011]; 2011.

Jackson C, Bortz A, Boneh D, Mitchell JC. Protecting browser state from web privacy attacks; ACM. WWW 06. p. 737–44.

Jang D, Jhala R, Lerner S, Shacham H. An empirical study of privacy-violating information flows in JavaScript web applications; ACM. CCS '10. p. 270–83.

Jim T, Swamy N, Hicks M. Defeating script injection attacks with browser-enforced embedded policies. In: Proceedings of the 16th international conference on World Wide Web. New York, NY, USA: ACM; WWW '07; 2007. p. 601–10.

JonDonym . Anonymity test. Web site: http://ip-check.info/?lang=en [Last accessed: 15 August 2011]; 2011a.

JonDonym . JonDoFox. Web site: http://anonymous-proxy-servers.net/en/jondofox.html [Last accessed: 15 August 2011]; 2011b.

Jung EH. FoxyProxy. Web site: http://getfoxyproxy.org [Last accessed: 15 August 2011]; 2011.

Karopoulos G, Kambourakis G, Gritzalis S, Konstantinou E. A framework for identity privacy in sip. Journal of Network and Computer Applications 2010;33(1):16–28.

KnowPrivacy . Web bugs. Web site: http://knowprivacy.org/web_bugs.html [Last accessed: 15 August 2011]; 2011.

Kpsell S. Low Latency Anonymous Communication How Long Are Users Willing to Wait?; Springer Berlin Heidelberg; volume 3995. p. 221–37.

Krishnamurthy B, Wills C. Privacy diffusion on the web: a longitudinal perspective; ACM. WWW '09. p. 541–50. ACM ID: 1526782.

Kubieziel J. To be or I2P. p. 1–4.

Lambrecht A, Tucker C. When does retargeting work? timing information specificity. SSRN eLibrary 2011;.

Langton A. Check Browser headers. Web site: http://andylangton.co.uk/online-tools/check-browser-headers [Last accessed: 15 August 2011]; 2011.

Lenhard J, Loesing K, Wirtz G. Performance Measurements of Tor Hidden Services in Low-Bandwidth Access Networks; Springer-Verlag. ACNS 09. p. 324–41.

Li B, Erdin E, Günes MH, Bebis G, Shipley T. An analysis of anonymity technology usage; Springer-Verlag. TMA'11. p. 108–21. ACM ID: 1986295.

Liberatore M, Levine BN. Inferring the source of encrypted HTTP connections; ACM. CCS '06. p. 255–63. ACM ID: 1180437.

Linn J. Technology and web user data privacy: A survey of risks and countermeasures. IEEE Security and Privacy 2005;3(1):52–8.

Loesing K, Sandmann W, Wilms C, Wirtz G. Performance Measurements and Statistics of Tor Hidden Services; IEEE Computer Society; volume 0. p. 1–7.

Maone G. NoScript. Web site: http://noscript.net [Last accessed: 15 August 2011]; 2011.

Margasiński I, Szczypiorski K. Web Privacy: an Essential Part of Electronic Commerce. p. 65–72.

Margasiński I, Szczypiorski K. VAST: versatile anonymous system for web users; Springer-Verlag. p. 71–82.

Marshall J. CGIProxy 2.1beta19 - HTTP/FTP Proxy in a CGI Script. Web site: http://www.jmarshall.com/tools/cgiproxy/ [Last accessed: 15 August 2011]; 2008.

Martin D, Schulman A. Deanonymizing Users of the SafeWeb Anonymizing Service; USENIX Association. p. 123–37.

Martin D, Wu H, Alsaid A. Hidden surveillance by web sites: Web bugs in contemporary use. Commun ACM 2003;46(12):258–64.

Mayer J, Narayanan A. DoNotTrack.Us. Web site: http://www.donottrack.us [Last accessed: 15 August 2011]; 2011.

Mayer J, Narayanan A, Stamn S. Do Not Track: A Universal Third-Party Web Tracking Opt Out. Internet-Draft draft-mayer-do-not-track-00; IETF Secretariat; 2011.

Mazieres D, Boneh D, Slack Q, Hamburg M, Bittau A, Handley M. Cryptographic protection of TCP Streams (tcpcrypt). Number draft-bittau-tcp-crypt-00, 2011.

Mccoy D, Bauer K, Grunwald D, Kohno T, Sicker D. Shining Light in Dark Places: Understanding the Tor Network; Springer-Verlag. PETS '08. p. 63–76.

Michael, Ares2, Erunno, Khrin and MonztA . Easy List. Web site: https://easylist.adblockplus.org [Last accessed: 15 August 2011]; 2011.

Microsoft Corporation . Flash Cookies. Web site: http://ie.microsoft.com/testdrive/Browser/FlashCookies/Default.html [Last accessed: 15 August 2011]; 2011.

Miyazaki AD. Online privacy and the disclosure of cookie use: Effects on consumer trust and anticipated patronage. Journal of Public Policy & Marketing 2008;27(1):19–33.

Mulazzani M, Huber M, Weippl ER. Anonymity and monitoring: how to monitor the infrastructure of an anonymity system. Trans Sys Man Cyber Part C 2010;40(5):539–46.

Multiproxy . Multiproxy. Web site: http://multiproxy.org [Last accessed: 15 August 2011]; 2001.

Murdoch SJ, Danezis G. Low-cost traffic analysis of Tor; IEEE. p. 183–95.

Murdoch SJ, Zieliński P. Sampled traffic analysis by internet-exchange-level adversaries. In: Borisov N, Golle P, editors. Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007). Ottawa, Canada: Springer; 2007. .

My-Proxy . Proxy List. Web site: http://proxies.my-proxy.com/ [Last accessed: 15 August 2011]; 2011.

Nentwich F, Jovanovic N, Kirda E, Kruegel C, Vigna G. Cross-site scripting prevention with dynamic data tainting and static analysis. In: Proceedings of the Network and Distributed System Security Symposium (NDSS'07). NDSS '07; 2007. .

Odvarko J. Firecookie. Web site: http://www.softwareishard.com/blog/firecookie/ [Last accessed: 15 August 2011]; 2011.

OptimizeGoogle . OptimizeGoogle. Web site: http://www.optimizegoogle.com [Last accessed: 15 August 2011]; 2011.

Palant W. Adblock Plus. Web site: http://adblockplus.org [Last accessed: 15 August 2011]; 2011.

Panchenko A, Pimenidis L, Renner J. Performance Analysis of Anonymous Communication Channels Provided by Tor; IEEE Computer Society. p. 221–8.

Peddinti ST, Saxena N. On the privacy of web search based on query obfuscation: a case study of TrackMeNot; Springer-Verlag. PETS'10. p. 19–37. ACM ID: 1881153.

Pfitzmann A, Hansen M. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. Web site: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf [Last accessed: 15 August 2011]; 2010. V0.34.

Privoxy . Privoxy - Home Page. Web site: http://www.privoxy.org [Last accessed: 15 August 2011]; 2010.

PublicProxyServers.com . Public Proxy Servers. Web site: http://www.publicproxyservers.com [Last accessed: 15 August 2011]; 2011.

Ragkhitwetsagul R. Web Bug Detector Implementing P3P Compact Policy for Mozilla Firefox, 2007.

Raymond JF. Traffic analysis: protocols, attacks, design issues, and open problems; Springer-Verlag New York, Inc. p. 10–29.

Reed MG, Syverson PF, Goldschlag DM. Anonymous connections and onion routing. IEEE Journal on Selected Areas in Communications 1998;16(4):482–94.

Ren J, Wu J. Survey on anonymous communications in computer networks. Computer Communications 2010;33:420–31. ACM ID: 1710240.

Rezgui A, Bouguettaya A, Eltoweissy MY. Privacy on the web: Facts, challenges, and solutions. IEEE Security and Privacy 2003;1(6):40–9.

Rosinstrument.com . Free Public Proxy Server List. Web site: http://tools.rosinstrument.com/proxy/ [Last accessed: 15 August 2011]; 2011.

Saint-Jean F, Johnson A, Boneh D, Feigenbaum J. Private web search; ACM. WPES '07. p. 84–90. ACM ID: 1314351.

Schilling T. Cookie Monster 1.0.5. Web site: https://addons.mozilla.org/en-US/firefox/addon/cookie-monster/ [Last accessed: 15 August 2011]; 2011.

Schlegel R, Wong DS. Low latency high bandwidth anonymous overlay network with anonymous routing. Cryptology ePrint Archive, Report 2009/294; 2009. http://eprint.iacr.org/. [Last ac-

cessed: 15 August 2011].

Schomburg J. Anonymity Techniques - Usability Tests of Major Anonymity Networks. p. 49–58.

Scroogle . Scroogle. Web site: https://www.scroogle.org [Last accessed: 15 August 2011]; 2011.

Senicar V, Jerman-Blazic B, Klobucar T. Privacy-enhancing technologies–approaches and development. Computer Standards & Interfaces 2003;25(2):147–58.

Serjantov A, Sewell P. Passive Attack Analysis for Connection-Based Anonymity Systems; Springer Berlin Heidelberg; volume 2808. p. 116–31.

Shankar U, Karlof C. Doppelganger: Better browser privacy without the bother; ACM. CCS 06. p. 154–67.

Showip . Ros instrument Whois. Web site: http://rosinstrument.com/cgi-bin/wi.pl [Last accessed: 15 August 2011]; 2011a.

Showip . showip. Web site: http://showip.net [Last accessed: 15 August 2011]; 2011b.

Shubina AM, Smith SW. Using caching for browsing anonymity. SIGecom Exch 2003;4:11–20.

Snader R, Borisov N. A tune-up for tor: Improving security and performance in the tor network. In: Network and Distributed System Security Symposium. NDSS 2008; 2008. .

Solipo . Solipo - Polipo GUI for Windows. Web site: http://serennz.sakura.ne.jp/toybox/solipo/ [Last accessed: 15 August 2011]; 2010.

Solomon M. PithHelmet. Web site: http://culater.net/software/PithHelmet/PithHelmet.php [Last accessed: 15 August 2011]; 2011.

Startingpage . Startingpage. Web site: http://startingpage.com [Last accessed: 01 October 2011]; 2011.

StatCounter . StatCounter Global Stats. Web site: http://gs.statcounter.com [Last accessed: 15 August 2011]; 2011.

Sun Q, Simon DR, Wang YM, Russell W, Padmanabhan VN, Qiu L. Statistical identification of encrypted web browsing traffic. IEEE SYMPOSIUM ON SECURITY AND PRIVACY 2002;.

Syverson P, Tsudik G, Reed M, Landwehr C. Towards an analysis of onion routing security; Springer-Verlag New York, Inc. p. 96–114.

Thaler D. JavaScript Blacklist. Web site: http://homepage.mac.com/drewthaler/jsblacklist/ [Last accessed: 15 August 2011]; 2011.

The Tor project, Inc . Tor - Anonymity Online. Web site: https://www.torproject.org [Last accessed: 15 August 2011]; 2011a.

The Tor project, Inc . Tor Browser Bundle. Web site: https://www.torproject.org/projects/torbrowser.html.en [Last accessed: 15 August 2011]; 2011b.

The Tor project, Inc . Torbutton. Web site: https://www.torproject.org/projects/vidalia.html.en [Last accessed: 15 August 2011]; 2011c.

The Tor project, Inc . Vidalia. Web site: https://www.torproject.org/projects/vidalia.html.en [Last accessed: 15 August 2011]; 2011d.

TOR FAQ . Why do we need Polipo or Privoxy with Tor? Which is better? Web site: https://trac.torproject.org/projects/tor/wiki/doc/TorFAQ [Last accessed: 15 August 2011]; 2011.

Toubiana V, Narayanan A, Boneh D, Nissenbaum H, Barocas S. Adnostic: Privacy Preserving Targeted Advertising; Internet Society (ISOC).

V@no . Cookies Manager+ 1.5.1. Web site: https://addons.mozilla.org/en-US/firefox/addon/cookies-manager-plus/ [Last accessed: 15 August 2011]; 2011.

WebAttack, Inc . PopupTest.com. Web site: http://popuptest.com [Last accessed: 15 August 2011]; 2011.

Weisbein J. Interview with Michael Gundlach (AdBlock For Chrome/Safari Developer). Web site: http://www.besttechie.net/2011/02/16/interview-with-michael-gundlach-adblock-developer/ [Last accessed: 15 August 2011]; 2011.

Wendolsky R, Herrmann D, Federrath H. Performance comparison of low-latency anonymisation services from a user perspective; Springer-Verlag. PET07. p. 233–53.

Westermann B, Kesdogan D. Malice versus an.on: Possible risks of missing replay and integrity protection. In: Proceedings of Financial Cryptography and Data Security (FC'11). 2011. .

Westermann B, Wendolsky R, Pimenidis L, Kesdogan D. Cryptographic Protocol Analysis of AN.ON; Springer Berlin Heidelberg; volume 6052. p. 114–28.

Yamada A, Hara M, Miyake Y. Web tracking site detection based on temporal link analysis and automatic blacklist generation. Journal of Information Processing 2011;51(10):62–73.

Yamada A, Masanori H, Miyake Y. Web Tracking Site Detection Based on Temporal Link Analysis; IEEE Computer Society; volume 0. p. 626–31.

Yamaoka D. CookieCuller 1.4. Web site: http://cookieculler.mozdev.org/ [Last accessed: 15 August 2011]; 2011.

Yan J, Liu N, Wang G, Zhang W, Jiang Y, Chen Z. How much can behavioral targeting help online advertising?; ACM. WWW 09. p. 261–70.

Yardley G. BetterPrivacy. Web site: http://netticat.ath.cx/BetterPrivacy/BetterPrivacy.htm [Last accessed: 15 August 2011]; 2011.

Yu D, Chander A, Islam N, Serikov I. Javascript instrumentation for browser security. SIGPLAN Not 2007;42(1):237–49.

Yue C, Wang H. Profit-aware overload protection in e-commerce web sites. Journal of Network and Computer Applications 2009;32(2):347–56.

Yue C, Xie M, Wang H. Automatic Cookie Usage Setting with CookiePicker; IEEE. p. 460–70.

Yue C, Xie M, Wang H. An automatic http cookie management system. Computer Networks 2010;54(13):2182–98.

Zalewski M. Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks. 1st ed. No Starch Press, 2005.

Zalewski M. the new p0f. Web site: http://lcamtuf.coredump.cx/p0f/p0f.shtml [Last accessed: 15 August 2011]; 2006.

Zantout B, Haraty R. I2P Data Communication System; IARIA. p. 401–9.

zzz , Schimmer L. Peer Profiling and Selection in the I2P Anonymous Network.