Private Wars in Secret Life

Igor Margasiński, Krzysztof Szczypiorski, Krzysztof M. Brzeziński
Warsaw University of Technology, Institute of Telecommunications
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland

Abstract—**Intrusions into the privacy of Web users, launched on a massive scale mostly to get commercial advantage, are not only annoying, but also start to hinder the further development of the Web and its accepted commercial uses. In this paper we seek the efficient ways of implementing privacy by assuring *anonymity* understood as *unlinkability* – the impossibility to reliably correlate the visible communication with a particular user. We briefly survey the state of the art and identify the deficiencies of existing anonymity solutions. We then submit the original approach to anonymity assurance: the VAST [12] system, which is based on a particular unlinkability notion – the impossibility to reliably correlate communication with the intentional behaviour of a particular user.**

Index terms — web privacy, privacy enhancing technologies, network attacks

I. INTRODUCTION

The abuse of end user privacy, in its sociological, financial, and technical aspects, is quickly becoming one of the most important topics in Web security. Obtaining, collating and using of personal information, mostly to gain commercial advantage, constitute a major and global attack on Web users. Privacy intrusion is intentional - personally attributable information is a "hot" commodity.

According to a recent survey [10], most customers do not trust companies to handle their personal information properly. Users are looking for solutions that can assure the protection of their privacy. They require tools and solutions that would grant them control over information about themselves that is potentially available for third parties to pick. The survey further reveals, for example, that having company privacy practices verified by a third party would lead 91% customers to declare that they would do more business with such a company. More than half the customers claim that, were they confident that a company really follows its privacy policies, they would be likely to recommend that company to friends and family. The Canadian report [6] is even more pronounced. It found that 83% of consumers who had never shopped on-line claimed that their reluctance was due to not knowing what was being done with their information and who was watching their surfing habits. 69% of frequent Internet purchasers say they have concerns about handing out personal information like credit card numbers on-line.

Privacy abuse on the Web has many facets. Apart from a simple theft (such as stealing the credit card numbers), much more subtle attacks are being launched, in particular – consisting in compiling user *profiles* (personally attributable information on user's habits, preferences, interests, etc.). Irrespective of the kind and immediate outcome of the attack, the long-term results are user discomfort, ranging from mild irritation up to the total denial of Web use, and commercial marginalization of Internet as a whole (this seems a rather far-fetched claim, but in our opinion it is more likely than one is ready to admit). These serious consequences call for effective countermeasures.

"Full" privacy, as the ability to withhold any personal information during Web browsing, is technically impossible to attain. User activity is, by definition, visible e.g. to intermediate nodes and WWW servers (even in the absence of "leaks" – taps in communication links). Therefore, we will seek the efficient ways of implementing privacy by assuring **anonymity**, understood as **unlinkability**, or impossibility to reliably correlate the visible communication with a particular user.

In this paper we will survey the existing tools and (partial) solutions of the privacy/anonymity problem that are conceptually and technically specific to WWW browsing (as opposed to, e.g., e-mail communication). Basing on the identification of their deficiencies, we will propose the original approach to anonymity assurance that, we claim, is inherently more powerful than current tools and, arguably, resistant to currently known attack patterns. The proposed system **VAST** [12] (Versatile Anonymous System for Web Users) is based on a particular unlinkability notion, namely, the impossibility to reliably correlate the visible communication with the **intended behaviour** of a particular user.

II. PRIVACY RISKS

The World Wide Web (WWW), in its present shape, does not provide adequate privacy protection. A vulnerability gap stays in place, or even widens, despite evident progress both in Internet technology and electronic commerce. Large specialized companies are the major drivers in utilization of this technology gap to their own commercial advantage. The privacy risks can be examined from a number of different angles. They can be roughly divided into internal, communication link originated and Web server originated.

A. Internal Risks

Internal security may be understood as security in the immediate environment of the user. Here, the main problem is access to user's personal data and Web activity information (e.g., a log of visited websites), kept on his own workstation, by local network administrators, employers or other third parties like Internet service providers. What is worrying, the URL (*Uniform Resource Locator*) addresses can inform not only about the sites visited by a user, but also about the way he filled out HTML (*HyperText Markup Language*) forms. This can be accomplished by means of the method GET of HTTP (*HyperText Transfer Protocol*) protocol [2].

An important privacy violation risk in relation to, but not necessarily during, web browsing, comes from the possibility of planting **malware**, e.g. a *Trojan horse*, on a client machine. Such programs can scan the machine and send to a

predetermined recipient any locally stored information, including personal information and the history of previous web visits.

B. Communication Link Related Risks

The basic technique of attacks concentrating on communication links is **sniffing** – copying the transported data by means of a suitable device (e.g., in a local environment – a network interface operating in a "promiscuous mode"). The main protocol of WWW is HTTP. Its security is derived from the security of the TCP/IP (*Transmission Control Protocol / Internet Protocol*) protocol suite, which does not account for the possibility of sniffing attacks. A user of a Web browser should be aware that information about visited sites and data from completed forms can be easily accessed also by other Internet users.

According to Harris Interactive report [10], 70% of customers claim that their major concern about online shopping is that Web transactions may not be secure and 69% that hackers could steal their personal data.

C. Web Server Originated Risks

A website may obtain a wide range of information about a client. A server gets a client's IP address while establishing the connection. The origin of each individual request and its association with each host are known to a server. A standard practice for most websites is to log HTTP requests with this information to their Web server. Owners of a website have also access to date of request, duration, user's name – HTTP identification, information about errors of HTTP transaction, referrer link, and user-agent information.

Sending a URL address of a previously visited page (a *referrer link*) to a server constitutes a major privacy violation. The HTTP specification says that the availability of information about a referrer link should be optional, but not a single existing user agent did in fact incorporate the possibility of turning off this mechanism. The significance of this violation is increased by the fact that the URI (*Uniform Resource Identifier*) address may contain data from HTML forms. An URI string may, in particular, contain keywords introduced into Web search engines.

The next mechanism that can be used to take privacy away from users is state management by means of **Cookies** [11]. This technique was introduced to allow, in a stateless protocol like HTTP, for the customization of services provided to different users according to the history of their previous visits at a server. However, the Cookie mechanism has acquired new uses, not intended by its creators. This mechanism is now often used to create and enlarge databases with detailed users' profiles. The statistics [10] show that the main concern for 75% of customers is that web companies they take their custom to will provide their personal information to other companies without their permission.

III. CURRENT SOLUTIONS TO PRIVACY VIOLATIONS

The space of partial solutions to the problem of privacy violations can be decomposed into a number of viewpoints, with complementary (rather than orthogonal and disjoint)

classes of mechanisms in each of the viewpoints. For further discussion we stick to two basic viewpoints:

- architectural viewpoint:
    - client-side and network-based solutions;

- protection policy viewpoint:
    - negotiation of the declared level of privacy protection, based on a certain level of mutual trust;
    - privacy protection based on the assumption of untrustworthy and hostile environment.

A. Client-side Utilities

Solutions of this type can play only a secondary role. The functions of protection mechanisms running on a client machine are: monitoring and control of all connections to and from a user's computer (i.e., a *personal firewall*), the management of the Cookies mechanism, system cleaning (removal of history or cookies files), blocking banner ads and detecting *Trojan horses*. A growing number of such applications combine many individual functions. However, they are not able to e.g. protect (conceal) data that may identify a host, like an IP address.

B. Network Solutions - Third Party Proxy Servers

A **Proxy** acts as a middleman in the process of Web browsing. Adding a proxy between a user and a Web server allows the hiding of all kinds of information about a user from a destination Web server. A server can only access information about a proxy. In addition, a secure connection can be established between a user agent and a proxy, using the SSL/TLS protocol (*Secure Socket Layer / Transport Layer Security*) [5]. If such a connection is used, other parties, such as ISP (*Internet Service Provider*), a LAN (*Local Area Network*) administrator, or just eavesdroppers cannot access the transferred information. Another advantage of a proxy is the ease of control and filtration of the transferred content. A proxy also allows for the management of the Cookie mechanism, blocking of unasked for, annoying or dangerous "extras" (popup windows, banner ads etc.), and also deleting client-side scripts or programs.

Using a proxy to *anonymize* Internet services is a widely known and popular technique, due to the high efficiency of hiding users' identity data, easy access to the service, no additional requirements imposed upon users (only an Internet access and a standard Web browser is needed), simple architecture, insignificant delays for Web navigation, relatively low cost of system implementation, no need for the modification of existing nodes and protocols.

The use of Third Party Proxy Servers has also serious disadvantages. Currently employed proxy servers have access to information about the Web activity of the users. Anonymity service providers induce the belief that this data is not collected, used or shared. However, if the anonymity service provider for any reason breaks this declaration, the user will be exposed to an even greater risk than in case of traditional Web browsing, because information collected by different websites is much more difficult to combine and profile. Furthermore, proxy servers do not protect against attacks by traffic analysis.

An eavesdropper (i.e. a third party) can observe the volume of transmitted data and correlate inputs and outputs (proxy server requests). Proxy servers also limit the sets of elements which can be downloaded. Some of the HTML standard enhancements (like JavaScript) can create high risks for the whole system. It is possible and relatively straightforward to perform attacks using similar technologies. Such attacks can completely compromise a proxy server system.

C. Network Solutions - Chaining with Encryption

One of the directions for improving the overall performance (especially the security level provided) of a proxy-based systems has been to replace a single proxy with a network of many intermediate nodes, where packets are routed at random through this network, and each node encrypts packets with a different public key. The identity of both a sender and a receiver is never disclosed to any single proxy in a network, and due to random routing of encrypted packets an attack based on traffic analysis is unlikely to succeed. This idea comes from David Chaum and has been originally devised to provide anonymity for electronic main (the MIXNET system [3]).

More or less successful adaptations of the concept to WWW systems have already been implemented by a few organizations (*Onion Routing* [9], *Crowds* [14], *Freedom* [8]). To encrypt and decrypt exchanged packets, these systems use additional software executed on a user's computer. This software takes over all communication with Internet and rerouting of packets to service servers. Usually the systems also provide anonymous access to other services such as e-mail, news, and file sharing.

The popularity of this type of systems is limited. In reality, systems based on chaining with encryption do not manage to eliminate all risks of traffic analysis attacks. Anonymity still depends on a third party – information about user's Web activity is dispersed between many proxies, but there can be no guarantee that proxies do not collaborate with each other. Each proxy should belong to a different infrastructure provider, but in the past many examples of cooperation between independent companies in the process of tracking Web users have been revealed. Why would it be any different in this case?

Systems based on a network of proxies require an expensive infrastructure, which discourages potential investors. Serious delays, which are acceptable for e-mail, are a serious obstacle for Web browsing. To increase the system performance, it is necessary to employ powerful and fast (and thus expensive) computers as proxies.

D. Privacy Negotiation - P3P Protocol

The Platform for Privacy Preferences Protocol (P3P) [13] is not a privacy protection mechanism *per se*, but rather a tool allowing a user to be informed of potential privacy risks, and to decide if his interest in contacting a given service is worth taking the risk. P3P introduces a uniform and machine-readable format for website **privacy policies** and for user's private data collected by his Web browser. A user's browser can read this information and automatically decide, e.g., whether to send user's id information or to allow Cookies. P3P is only a mechanism; it does not guarantee in any way the advertised privacy policy of a website to be true. However, as a standardized mechanism, it may be attractive to web companies in the long term, as it can increase mutual trust (and thus – the volume of electronic trade). P3P can help in achieving harmony between companies' economical needs for information (which is required to provide services) and customers' rights to privacy and control over their personal information

The recent Web browsers are equipped with a P3P. However, up till now only very few Web pages have supported the P3P technology.

IV. THE VAST SYSTEM

The goal has been to design a privacy protection method tailored specifically for the WWW system. Such system, called VAST [12] (Versatile Anonymous System for Web Users) is the subject of the remaining part of the paper.

A. The idea

VAST is a "zero-trust" method, which assumes a hostile and untrustworthy environment. It is thus complementary to negotiation-based approaches such as P3P. VAST is also an *active* method – in some aspects its operation resembles an attack (classified in [1] as *subterfuge* or *deflection*) upon a party trying to profile a user. This feature, by some considered as "ethically dubious", is a price to pay for the effectiveness of the method, and we claim that this approach is justified by the volume of privacy violations and harm inflicted by them on the users.

Other known privacy protection systems are based on *hiding* (e.g., substituting, encrypting) or *masking* (e.g., dispersing) the user traffic. When information is eventually revealed (e.g., decrypted) or unmasked, the protection is gone. The idea of VAST is to introduce a dummy traffic that *could have originated* from the user, but actually does not. Therefore, the observed traffic, regardless of whether eavesdropped, inferred from traffic analysis or just recorded by a network node, cannot be reliably attributed to a given user, and thus is useless for the attacker.

The design objectives of VAST have been:

- preservation of advantages of a "single third party proxy server" approach,

- providing versatile anonymity (with regard to all parties concerned),

- retention of speed (minimizing of performance differences between VAST and traditional browsing),

- accessibility – no additional requirements imposed upon users (this design objective has not been fully realized),

- facility of "real-life" implementation.
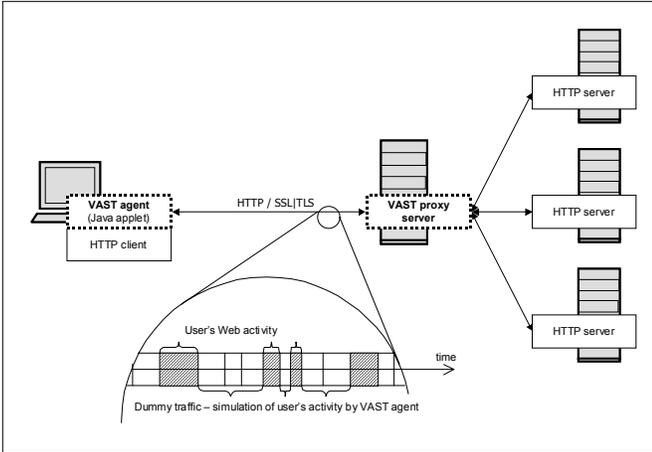
B. The architecture

Figure 1.   Architecture and operation of VAST

VAST introduces only one additional node – a kind of a proxy server. To achieve anonymity w.r.t. this server, and also to render useless traffic analysis attacks, a **specific kind of a *dummy traffic* generation mechanism is placed between a distant proxy and a local agent**. More web pages than actually requested by a user are transferred from a proxy to a client. Information about which content is actually the object of interest to the user rests with the user himself. An agent (a JAVA applet) cooperates with the user's browser. While the user reads the page contents, the agent simulates web activity of this user, by requesting random Websites from the proxy. This idea originates from the observation of a typical Web navigation behaviour. A user does not request Websites at all times. Requests are sent in various time intervals, after the user has read the contents. VAST, unlike other existing solutions, does not conduct any additional major activity *while* the transaction is taking place, but it rather utilizes free time in between, inherent in web browsing.

VAST consists of two key elements: an **agent** (a Java applet) placed in the Web browser environment, and a **proxy** placed between the agent and a destination Web server.

The primary functions of the VAST agent include: communication with the proxy server by means of the secure SSL/TLS protocol (*Secure Socket Layer* / *Transport Layer Security*) simulation of user Web activity; generation of URL (*Uniform Resource Locator*) addresses as a background for addresses requested by the user; receiving configuration parameters from the user and transmitting them to the proxy; requesting pages selected by the user and pages selected by the dummy traffic generator; receiving resources from the proxy; dividing resources into a group of pages chosen by the user and dummy pages; presentation of pages chosen by the user (skipping the dummy pages); analysis of the level of user anonymity, calculated as a proportion of resources downloaded by the user to resources downloaded by the dummy traffic generator (which serves as a simulator of user activity); presentation of actual anonymity level to the user via a graphic interface.

The VAST proxy server is very similar to popular anonymous proxy systems. The main difference is the absence of a user interface. This function was moved to the VAST agent. The primary functions of the VAST proxy server are: hiding all user-identifiable data (including IP addresses) from a destination Web server; encrypting all data transmitted between the VAST agent and the VAST proxy (including the URL addresses of resources); optionally - encrypting all communication between the VAST proxy and a destination Web server; blocking cookies, scripts, programs and Java applets sent from destination Web servers.

C. Operation of VAST

For the purposes of this paper the following two terms are introduced:

- Web transaction – a series of HTTP client requests and corresponding server responses, which represent a single Web page,

- Subject session – a collection of Web transactions generated by a user, where all transactions can be connected with each other by links from transaction pages. In the rest of this paper a shorter name – session – will be used.

We presume that a potential eavesdropper, who has access to transmitted data, is able to extract individual transactions and sessions from observed communication. The dummy traffic generation corresponds to the establishment of additional sessions. Transactions which belong to these sessions typically take place while the user is reading the content of already received pages. The Agent also generates dummy traffic requests assigned to the original user session. This makes impossible any reliable distinction of a "true" user session. Specific properties of sessions generated by a human – identifiable semantic relations between transactions – are then lost. When the user starts a new session, the agent also restarts dummy sessions. An eavesdropper (who knows the algorithm of agent applet, which is open source), can not distinguish if a particular request comes from the user or from the simulator. The anonymity service provider – the strongest possible attacker – is only able to separate particular sessions. The provider may assume that one of these sessions is of interest to the user, but the he does not know which one it is. The provider also does not know which requests from particular session come from the user. The user can configure the number of dummy sessions. Given the bandwidth of an Internet connection and the frequency of requests, one can select the appropriate level of anonymity, which can be represented by the probability (P) that the user is interested in the subject of selected session.

$$P \leq \frac{1}{Number\ of\ dummy\ sessions\ +\ 1} \qquad (1)$$

It is important to note that a single dummy session provides anonymity called *probable innocence*. The user should configure the system before the first use – it is necessary to input a list of search engines preferred by the user. The Agent will then employ these engines to generate dummy traffic. The

Agent will use a **dictionary** of queries downloaded from the VAST proxy server. It is important for the dictionary to contain a large number of queries. If the user enters a query which is not in the dictionary, the VAST Agent will issue a warning – in this case the VAST service provider may be able to infer that the query was not generated by the simulator. A request for a page via a search engine marks the beginning of a new session. The same rules apply to the beginning of dummy sessions. At first, the user requests are not submitted immediately. The choice as to which transaction is executed first is made randomly. In subsequent transactions, user requests have priority over simulator requests. However, if their frequency is higher than the frequency of dummy transactions, the user gets an appropriate warning.

The graphic representation of a sample communication between the VAST Agent and the VAST proxy server is shown in figure 2. In this example, two dummy sessions were used. Cuboids represent WWW transactions in particular sessions. The arrows point to the user transactions.
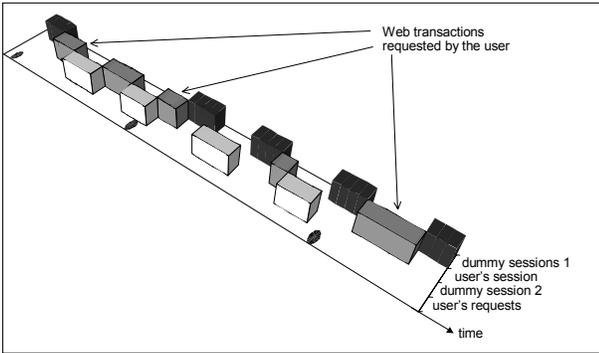

Figure 2. Sample communication the VAST agent and VAST proxy.

V. PERFORMANCE ISSUES

A. Quantitative aspects – user perspective

The presented VAST system is designed to provide high performance, as perceived by the user – similar to the performance of traditional Web browsing. Additional operations focused on providing anonymity occur when the user is reading the page contents. A question remains: how does the dummy traffic actually delay browsing? Sometimes the user just glances at the page. How long will he be forced to wait for the next page? VAST may block (disable) data, including banner ads, which originate from third party servers (i.e. AdServers). Statistical analysis conducted by the authors showed that the size of ads published on most popular Web pages and Web portals often exceeds 50% of total page size. The number of requests necessary to download a page is often a multiple of requests to destination servers. In the VAST system, requests to third party servers may be replaced by dummy requests. Users often do allow for the downloading of numerous advertising elements. We are entitled to claim that the replacement of ads with dummy traffic, which provides privacy protection, would be also acceptable (and welcome).

The volume of dummy traffic should be maintained on a certain level, in order to perform effective masking of user activity. The number of transactions performed in respective sessions should be approximately equal. Let t_d be the average time of downloading of a single Webpage; t_f – the average time of reading a page; t_w – the average delay in Webpage downloading induced by the VAST system in comparison to a traditional proxy server; n – the number of dummy sessions. Then, the delay t_w can be described as follows:

$$t_w = 0.5 \ t_d \qquad \text{for } t_f \geq n \ t_d \qquad (2)$$

(average time required to finish the current transaction)

$$t_w = n \ t_d - t_f + 0.5 t_d \qquad \text{for } t_f < n \ t_d \qquad (3)$$

($n t_d$ of dummy transactions have to be performed to provide proper level of anonymity). Finally we obtain:

$$t_w = \frac{|n \ t_d - t_f| + (n+1) \ t_d - t_f}{2} \qquad (4)$$

Table 1 shows t_w delays as a function of average Webpage downloading time t_d and average reading time t_f, for $n = 1$ (A) and $n = 2$ (B) dummy sessions. The following results can be obtained for a typical downloading time $t_d = 8$ [s]:

TABLE I. DELAYS INTRODUCED BY VAST (T_w)

n	t_f [s]	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	t_w [s]	12	11	10	9	8	7	6	5	4	4	4	4	4	4	4	4	4	4	4	4	4
2	t_w [s]	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4	4	4

According to our expectations, these results show that acceptable delays (similar to delays present in traditional anonymous proxy server systems) occur when users spend some time ($t_f \approx n t_d$) familiarizing with the Webpage content.

B. Qualitative aspects - security

Another performance-related aspect of VAST is its security, or its ability to withstand arbitrary attacks aimed at linking a particular user with the observed activity.

Communication between the user's computer and the proxy server is secured by means of the SSL/TLS protocol. It means that third parties in the local environment do not have access to transmitted data. From this perspective, anonymity is accomplished by *hiding*. Security in this area depends on the strength of the SSL/TLS protocol itself and cryptographic algorithms it utilizes. Because both the Agent and the proxy server are implemented by the anonymity service provider, it is possible to choose suitably advanced cryptographic algorithms (i.e. SSL version 3).

Between the user transactions, dummy traffic is being generated. This constitutes a very effective barrier against traffic analysis attacks. As already mentioned above, communication between the agent and the proxy server is very effectively guarded against a *sniffing attack* and the possibility

of correlating requests and answers based on timing relations. For communication between the proxy server and a destination Web server, encryption is not a key. Eavesdropping of this data results only in the interception of information about the activity of the VAST proxy server. If the system is employed by many users, this information is virtually worthless. In the extreme case, when there is only one active user and the eavesdropper has the ability to intercept all requests realized by the proxy server, security from the point of view of other Internet users is the same as from the VAST service provider perspective (i.e., still satisfactory).
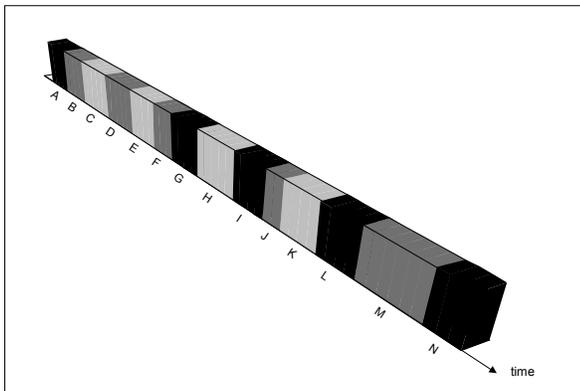

Figure 3. Communication between the agent and proxy server – from the proxy server's point of view (compare with fig. 2).

Let us now consider the VAST-provided anonymity from the perspective of a VAST service provider. Referring to figure 2, the attacker may only correlate particular requests such as shown in figure 3. We can observe that the proxy server may differentiate between particular sessions (three sessions distinguished in the picture with different shades of gray). The attacker may not determine which shades represent genuine user activity. He may also realize that user requests correspond to only some blocks represented by the same (unknown) shade. Therefore, it is possible, after conducting an analysis of transaction contents, to determine that data transmitted from sections: A, G, I, L, M is one session, B, D, F, J, M is the second, and C, E, H, K is the third. The potential attacker knows that the topic of one session is of interest to the user. He is able neither to determine which one it is nor to separate user requests (in this example: B, F, M).

We should also consider the attacks where the VAST service provider resends "fake" pages or a "fake" reference dictionary to find out if they have been requested by an applet or by a human being. In this case the attack can be successful. However, after this attack has been detected (which is relatively easy and unavoidable), the service is compromised and worthless to users. Therefore this attack can not be utilized to perform widespread profiling. The cost of the attack is higher than its profit.

From the destination Web server perspective, there is no ability to identify the system users. The only data available to the destination Web server concerns the proxy server, which

forwards user requests. Active elements placed on pages, which could communicate with the destination server, are removed by the VAST proxy server.

Until now, no implemented system provided effective protection against all the well known types of **traffic analysis attack** (like timing attacks, message volume attacks, flooding attacks, linking attacks). Systems based on the already mentioned idea of MIXNET do provide an effective protection against timing attacks or message volume attacks. Below we discuss protection offered by VAST against each of the attacks.

The **timing attack** consists in observing of communication events at the potential end points and searching for correlations between the beginning and/or the end of these (streams of) events in each possible end point. The VAST system provides the total protection against this type of attack, due to the specific dummy traffic generation mechanism. An eavesdropper is not able to differentiate between particular requests, because right after the finalization of one transaction the next one begins. Therefore, it is not possible to establish if a request belongs to a particular transaction. Obviously, in case where there is only one active system user, the eavesdropper can presume that all proxy requests come from this single user. However, even then, the anonymity of the user is not compromised and is the same as anonymity from the VAST proxy server perspective.

The **message volume attack** consists in using the observed transfer volume (i.e. message volume) as a criterion for correlating inputs with outputs. The VAST system fills periods of user inactivity with dummy traffic. However, the separation of particular messages from the encrypted transmission link between the agent and the proxy server is practically impossible.

The **flooding attack** is based on sending a large number of messages (flooding) or a stream of messages with certain characteristics. This is done in order to separate user messages. The VAST system protects against this type of attack. Even after the eventual isolation of a user message, it is still unknown which requests are generated by a machine and which come from a human.

VI. FUTURE IMPROVEMENTS

The VAST system, as presented above, does not offer an effective protection against the **linking attack** which is based on a long-term observation. This attack uses changes in traffic patterns (recurring patterns of requests) related to the presence or absence of a connection. In order to maintain simplicity, we did not take into consideration such type of risks. However, it is possible to enhance the VAST system and include a mechanism that does provide effective protection against long-term linking attack. To this end, a mechanism of registering recurring requests should be used in the agent program. This would allow the dummy traffic to simulate the user activity not only in the course of one session, but also in longer term. The Agent will record recurring user requests. The agent program does not share this data with other parties. Recurring requests are accompanied by dummy traffic, which also simulates user activity in the course of many sessions. We should consider the transformation of the agent Java applet into a Java program

which could be called a **local proxy**. This will allow saving files locally on the user's computer. In this way it will be possible to save the history of user activity. This will also permit the storage of dummy traffic files and their use as a cache memory. The user may choose a page already downloaded during a dummy transaction. This greatly increases the navigation speed.

VII. CONCLUSIONS

In this paper we have introduced an original method – VAST – which provides protection of Web user's privacy by implementing versatile anonymity. This solution evolved from popular single proxy systems. It is a comprehensive technique which overcomes weaknesses of existing systems such as: serious, noticeable delays; access of a service provider to private user data; and high costs of service implementation. One of the novel ideas ideas of VAST – the use of Web search engine resources to generate dummy traffic between the local agent and the distant proxy – may also be regarded as its weakness. For users who pay for the amount of downloaded data, it means higher costs. We should stress that the system can block advertisement elements originating from third party servers. This, in turn, means that the graphic files from third parties are "traded" for dummy traffic. As usual, there is a price to pay for anonymity. To preserve full security, the user can not start navigating from direct URLs, but only from queries input into popular search engines. The requested phrases should be included in the VAST dictionary, which can be quite *vast* indeed. This means that, in some cases, the user would have to take a moment to think how to change his request to find what he is really looking for. Anonymity from the perspective of the VAST service provider is accomplished through masking. The provider may only presume, with certain probability chosen by the user, that particular requests come from the user.

Absolute anonymity of web users, achieved by technical means, seems to be elusive and practically impossible. It is also unclear whether, in view of current global security concerns, it should be pursued at all. However, VAST seems the closest mechanism available to such absolute anonymity.

REFERENCES

[1] Axelsson, S. Intrusion Detection Systems: A Survey and Taxonomy, Dept. of Computer Engineering, Chalmers Univ. of Technology, TR:99-15, March 2000

[2] Berners-Lee, T., Fielding, R., Frystyk, H.: Hypertext Transfer Protocol – HTTP/1.0. RFC 1945, 1996.

[3] Chaum, D. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM, 1981.

[4] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marchall, M. The Platform for Preferences 1.0 (P3P 1.0) Specification, W3C Recommendation, 16 April 2002.

[5] Dierks T., Allen C. The TLS-Protocol Version 1.0. RFC 2246, 1999.

[6] Ferneyhough, C. Online Security and Privacy Concerns on the Increase in Canada, Ipsos-Reid, 2001.

[7] Fielding, R., Gettys, J.,Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners- Lee T. HyperText Transfer Protocol – HTTP/1.1. RFC 2616, 1999.

[8] Goldberg, I., Shostack, A. Freedom Network 1.0 Architecture and Protocols. Zero-Knowledge Systems. White Paper, 1999.

[9] Goldschlag, D. M., Reed, M. G., Syverson, P. F. Onion Routing for Anonymous and Private Internet Connections. Communications of the ACM, 1999.

[10] Krane, D., Light, L., Gravitch D. Privacy On and Off the Internet: What Consumers Want. Harris Interactive, 2002.

[11] Kristol, R., Montulli, L. HTTP State Management Mechanism. RFC 2965, 2000.

[12] Margasiński, I., Szczypiorski, K. VAST: Versatile Anonymous System for Web Users. In: Enhanced Methods in Computer Security, Biometric and Artificial Intelligence Systems - Springer-Verlag, 2004.

[13] Presler-Marshall, M. The Platform for Privacy Preferences 1.0 Deployment Guide, W3C Note, 10 May 2001.

[14] Reiter, M.K., Rubin, A.D. Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security, 1997.

[15] Syverson, P. F., Goldschlag, D. M., Reed, M. G. Anonymous Connections and Onion Routing. IEEE Symposium on Security and Privacy, 1997.