VAST: Versatile Anonymous System for Web Users

IGOR MARGASIŃSKI, KRZYSZTOF SZCZYPIORSKI
Warsaw University of Technology, Institute of Telecommunications
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: {I.Margasinski, K.Szczypiorski}@tele.pw.edu.pl

Abstract: This paper presents an original method of providing versatile anonymity for Web users – VAST. It includes an introduction to the current techniques of providing anonymity in WWW system, both popular Third Party Proxy Servers and enhanced systems based on Chaining with Encryption. Limitations of these systems are discussed. In Third Party Proxy Servers – concentration of personal Web activity data; in Chaining with Encryption – low performance and high costs of network realization. Both classes of solutions do not eliminate all the risks of traffic analysis. The new method described – VAST – overcomes mentioned weaknesses and provides versatile anonymity for all parties involved in data exchange based on the WWW system. In this paper we also introduce a draft of the method implementation in Java language.

Key words: anonymity, privacy, anonymous web browsing, privacy-enhancing technology

1. INTRODUCTION

When leading their lives into the digital dimension, people have the right not to change their behaviours, and ways of fulfilling their needs. The moulding of our world, the creation of new environments of existence should belong to men not machines. The World Wide Web, which along with the e-mail is the most popular Internet application, lacks privacy protection. Privacy in the Web can be achieved using tools which provide **anonymity**. It is one of the ways to create a digital environment similar to our reality.

We can distinguish two kinds of anonymity: **person anonymity** and **message anonymity**. Person anonymity may be classified into: **sender anonymity** and **receiver anonymity**. It is of significance that the term “anonymity” is usually associated with some particular **point of view**. For example, anonymity of a poet generally means withholding his identity only from his readers and not from his publisher. The protection can be achieved by **hiding** (an access to crucial data is protected) or **masking** (the possibility to distinguish crucial data is protected – often by masking in so called **dummy traffic** – fake data). We should also mention the term “**unlinkability**”, which means that it is known that both sender and receiver participate in some communication, but they cannot be identified as communicating with each other. In this paper we concentrate on **receiver anonymity from all parties’ point of view**. Our aim is not to hide identity of user (receiver) from the anonymity service provider, but to hide his Web activity. Therefore, what we have in mind is the unlinkability of user and destination Web servers. We introduce a solution to a problem of the lack of Web privacy – VAST – ***V**ersatile **A**nonymous **S**ys**T**em for Web Users*.

2. RELATED WORK

Third Party Proxy Servers providing anonymity for Web users are gaining more and more popularity. Proxy is a distant machine – a middleman between a client and a server, which forwards user’s requests to destination Website and resends its content. This architecture creates the possibility of hiding all information about client from server (for example IP address). Additionally, it is possible to encrypt transmission between client and proxy server. Then all user Web activity will be hidden from parties who have an access to transferred Web transactions (for example from Internet Service Providers or from LAN administrators). Third Party Proxy Server also provides wide range of possibilities of control of transferred resources: it is likely to control HTTP State Management Mechanism – *Cookies* [8] and to block discarded additional elements (for example pop-up windows), and also to remove scripts and programs. Proxy can perform any filtering of transmitted documents. The advantages of proxies, responsible for their high popularity are: filling the technical loop-hole in WWW system related to lack of user’s privacy protection [7], high efficiency in hiding user’s identity data, easy access to the service, supported by no additional requirements from users (only an Internet access and standard Web browser needed), simple usage, insignificant delays of Web navigation, simplicity, relatively low costs required for system realization and no demands to modify existing network nodes and protocols – system implementation based on current well known standards. However, there are also serious disadvantages to Third Party Proxy Servers. Currently employed proxy servers have access to information about user’s Web activity. Anonymity service providers induce the belief that this data is not collected, used or shared. The user has to face the risk of concentration of personal Web activity in one place. If the anonymity service provider attempts to exercise this possibility, the user will be exposed to an even greater risk than in traditional Web browsing, because information collected by different Websites is much more difficult to put together. Furthermore, proxy servers do not protect against tracking by traffic analysis. An eavesdropper can observe the volume of transmitted data and correlate inputs and outputs (proxy server request). A very serious risk also exists here, because it is possible for third parities to track and profile users. The next disadvantage of proxy servers is the limitation of sets of elements which can be downloaded. Some of HTML standard enhancements (like JavaScript) can create high risks for the whole system [9]. It is possible and easy to perform powerful attacks using these technologies. The mentioned attacks can completely compromise proxy sever systems.

Having security limitations of single proxy servers in mind, the natural way to improve these systems is to disperse localization of information about user's Web activity. This is being done by replacing one server with many nodes. Each of these proxies has a partial knowledge about transferred data – it can be realized by public key cryptography – **Chaining with Encryption**. By repeatedly encrypting a message (for example, request to Web server) with public keys of succeeding nodes, we can transmit data from user's computer to destination server without disclosing the identity of both sender and receiver to every proxy. Routing among accessible nodes should be random. Thanks to this, packets alternate mutually, which in effect leads to an elimination of traffic analysis attack. This idea dates back to David Chaum's theory [2] of providing anonymity for electronic mail – MIXNET, where the enhanced proxy servers are the so-called MIXes. Their name has its origin in their characteristic role which is the mixing of delivered items. Before a MIX sends a message, it waits until it receives a batch of messages. Then it sends the messages outputting them in random order. Examples of systems based on MIXNET include: *Onion Routing* ([6], [11]), *Crowds* [10], *Freedom* [5] (first profitable system of this kind). Systems based on a network of many proxies require construction of expensive infrastructure. This brings in a discouraging factor for potential investors. The use of network of proxies is a great technique for providing anonymity during e-mail correspondence. However, serious delays, which are not inconvenient in sending e-mails, are a serious obstacle in Web browsing. To increase system's performance speed, it is necessary to employ powerful and fast computers as proxies. Yet, they are expensive. The authors of *Crowds* project presented an analysis of delays caused by sequent proxies of their system. We employ this data to illustrate the increase in time needed to receive a page – as a function of number of proxies. Presented results (table 1) show that the cost of providing anonymity in systems based on a network of proxies is a significant increase of waiting time. Moreover, Crowds system contains serious simplifications in comparison to MIXNET – which role is to increase the performance speed. In spite of these costly – from the security point of view – differences, performance is not satisfactory.

Number of proxies	Page size [kB]					
	0	1	2	3	4	5
2 + 1	20,8%	35,0%	23,7%	13,7%	17,8%	26,3%
2 + 2	42,1%	43,4%	32,3%	22,8%	18,6%	28,2%
2 + 3	73,1%	72,1%	60,7%	40,0%	36,6%	45,0%

Tab. 1. Delays as a function of number of proxies and page size in Crowds system (0% delay is for two proxies)

To accomplish encryption and decryption of exchanged data these systems use additional client-side software applications. This software takes over all communication with the Internet and reroutes packets to anonymous service servers. But a system which requires that a user installs additional software decreases his trust. Downloaded applications able to communicate with a public network may constitute so-called Trojan Horses. The user does not have the ability to check what really goes to the service server. This requirement also narrows the group of users, because it means dependency on a specific hardware and operating system. Systems based on chaining with encryption do not eliminate all risks of traffic analysis attacks. Anonymity is still dependent on a third party – information about user's Web activity was only dispersed between many proxies. However, there is no certainty that proxies do not collaborate with each other. In our opinion, WWW service requires an individual approach. When designing a system providing anonymity for Web browsing, concentrating on different aspects and parameters unlike in e-mail, should proof beneficial. Today more than ever, the WWW service increases its multimedia character. The users expect texts and graphics to be available immediately. Today's solutions though, still require very high financial investments. What is more, anonymity service provider cannot submit a proof that system proxies do not collaborate. Originators usually omit this fact and only mention curtly that each of proxies should belong to a different infrastructure provider. Is this a reason enough to support a belief that proxies do not collaborate? Nowadays we can see many examples of cooperation between independent companies in the process of tracking Web users and profiling them. Why would it be any different in this case? Implementation of this class of methods is not widely spread. We can only presume that in case of their popularization and implementation of proxies by many different companies, this possibility could be misused.

3. VAST OVERVIEW

Our goal was to design a method dedicated to WWW system specifics, which can take advantage of them. We have set the following principles for our solution:

- preservation of all advantages of single third party proxy servers,
- providing versatile anonymity (including service provider and also preventing risks of traffic analysis attacks),
- retention of speed (minimalization of performance differences between VAST usage and traditional browsing),
- accessibility – no additional requirements from users,
- facility to implement outside laboratories – relatively low costs.

VAST system contains only one proxy node. To achieve anonymity from proxy server's point of view and also to disable traffic analysis attack we are placing **specific kind of dummy traffic generation mechanism between distant proxy and local agent**. More pages than actually requested by user are transferred from proxy to client. Information about which content is the object of interest to the user stays with him. An agent (JAVA applet) cooperates with user's browser. At the time when the user is familiarizing himself with the page content, the agent simulates users Web activity – by requesting random Websites from proxy. The basis of this solution is the observation of a typical Web navigation. User does not request Websites at all times. Requests get sent in various time intervals and in between them, the user reads the content. VAST utilizes this fact. Additionally, VAST takes advantage of the simplicity of generating dummy traffic in WWW system – we mean that a wide range of Internet resources is indexed in Web search engines. We can simulate user's activity using it. The VAST system, unlike other existing solutions, does not conduct additional major activity while the transaction takes place, but it utilizes free time to take actions providing versatile anonymity. A source code of agent applet would be available to all. Then users would be able to check if it is not a *trojan horse*.

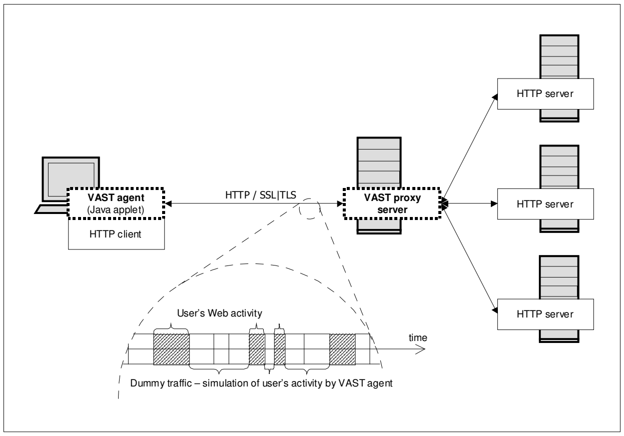

Fig. 1. VAST system scheme

VAST consists of two primary elements: **agent** placed in Web browser environment and **proxy** placed between the agent and a destination Web server.

Agent is an Java applet residing in user's Web browser. Primary functions of VAST agent include: communication with proxy server secured by SSL/TLS protocol (Secure Socket Layer / Transport Layer Security [3]), simulation of user Web activity, generation of URL (Uniform Resource Locator) addresses as a background for addresses requested by user, receiving configuration parameters from user and transmitting them to proxy, requesting pages selected by user and pages selected by simulator, receiving resources from proxy, dividing resources between group of pages chosen by user and dummy traffic pages, presentation of pages chosen by user (skipping dummy traffic pages), analysis of a level of user anonymity as a result of a proportion between resources downloaded by user and resources downloaded by simulator, presentation of actual anonymity level and communication with user by graphic interface.

Proxy server which is a part of the VAST system is very similar to popular anonymous proxy systems – the main difference is an absence of user interface. This function was moved to VAST agent. Primary functions of VAST agent are: hiding all user's identifiable data from destination Web server – IP address among others, encrypting all data transmitted between VAST agent and VAST proxy – resources' URL addresses among others, optional encrypting communication between VAST proxy and destination Web server, blocking cookies from destination Web server, blocking scripts and programs from destination Web server and blocking Java applets from destination Web server.

4. DESIGN AND IMPLEMENTATION

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

- *Web transaction* – a series of HTTP client requests and correspondent server responses, which represent a single Web page (HTML files and contained elements, i.e. graphic files),
- *Subject session* – collection of Web transactions generated by user – where all transactions can be connected with each other by links from transactions pages. In the rest of this paper a shorter name – *session* – will be used.

We presume that potential eavesdropper, who has an access to transmitted data is able to separate each transactions and sessions from communication. The dummy traffic generation complies with establishment of additional sessions. Transactions which belong to these sessions typically take place while the user is familiarizing himself with the content of pages already received. Agent generates dummy traffic requests assigned to user's session as well. Thanks to this, the reliable distinction as to which session comes from the user is not possible to make. Specific properties of session generated by human – thematic relations between transactions – are then lost. When user starts a new session, the agent also restarts dummy sessions. An eavesdropper (who knows the algorithm of agent applet – open source), can not distinguish if a particular request comes from user or from simulator. The anonymity service provider – the strongest possible attacker – is only able to separate particular sessions. The provider may know than that one of these sessions is of an interest to the user, but he does not know which one. The provider also does not know which requests from particular session come from the user. The user can configure the number of dummy sessions. Having the bandwidth of Internet connection and the frequency of requests in mind, one can select appropriate level of anonymity, which can be represented by the probability (*P*) of the fact that the user is interested in the subject of selected session.

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

We should mention here that conducting only one dummy session provides anonymity called *probable innocence* [10]. Closer analysis of VAST security is presented in section 5. The user will have to configure the system before he starts using it – this is necessary to input a list of search engines preferred by user. The agent will then employ them to generate dummy traffic. The agent will use a **dictionary** of queries downloaded from 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 inform about it and warn that the VAST service provider may infer that the query was not generated by the simulator. A request of a page trough a search engine means the beginning of a new session. The same rules apply to the beginning of dummy sessions. At first user's requests are not immediately ran. The choice which transaction is executed first is random. In subsequent transactions user's requests have priority of execution by an agent. However, if their frequency is higher than the frequency of dummy transactions, the user gets an appropriate message. The agent displays a warning, based on higher frequency of certain transactions, that eavesdropper may presume that requests came from the user. The graphic representation of a sample communication between VAST agent and 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's transactions.

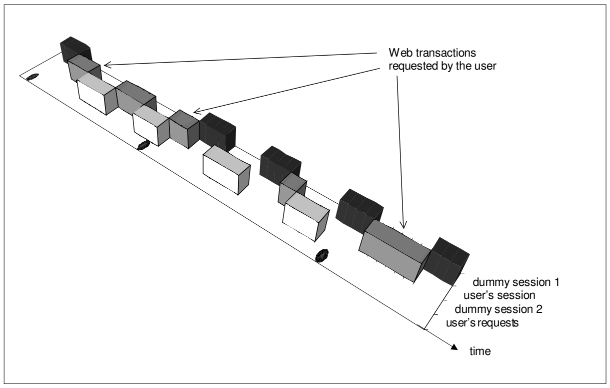

Fig. 2. Sample of communication between VAST agent and VAST proxy server. The cuboids represent single WWW transactions

Performance

The presented system had been designed to provide high performance – similar to performance in traditional Web browsing. Additional operations focused on providing anonymity neither take place at the time of realization of user's requests nor at the time of data downloading. They occur when the user is familiarizing himself with the requested page. A question remains: how does the dummy traffic actually delay browsing? Sometimes the user browses briefly through the page's content. How long will he be forced to wait? VAST may block data which originates in third party servers (i.e. AdServers). It means the disabling of all third party banner ads often published on various Web pages. The statistics analysis conducted by authors showed that 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 pages is often a multiple number of requests to destination server. In the VAST system requests to the third party servers may be replaced by dummy requests. Users allow downloading of numerous advertising elements. So we are justified in thinking that replacement of ads with dummy traffic which provide privacy protection would be also acceptable.

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

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

(average time required to finish current transaction)

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

(nt_d of dummy transactions have to be performed to provide proper level of anonymity). We can sum it up as:

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

Figure 3 shows t_w delays a function of average Webpage downloading time t_d and average time of user's familiarizing with content t_f, suitable 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]:

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

Tab. 2. Delays introduced by VAST (t_w) system as a function of user's familiarizing with Web content time

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

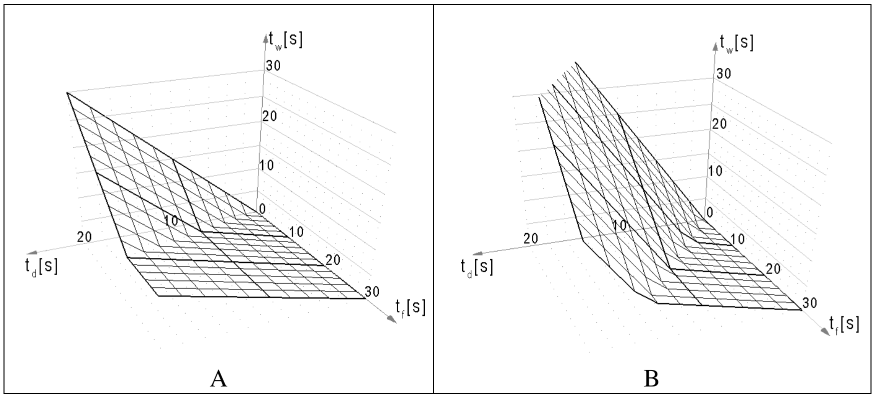

Fig. 3. Delays induced by VAST system in comparison to traditional proxy server (A – one dummy session, B – two dummy sessions)

5. SECURITY

The communication between user's computer and proxy server is secured by SSL/TLS protocol. It means that parties from the local environment do not have an access to transmitted data. Anonymity **from the local environment parties point of view** is accomplished through hiding. The security in this area is based on the strength of the SSL/TLS protocol itself and cryptographic algorithms utilized in it. Because both agent and proxy server are elements implemented by anonymity service provider, it is possible to choose appropriate cryptographic algorithms (i.e. SSL version 3). In the intervals between user's transactions dummy traffic is being generated. It constitutes a very effective barrier against traffic analysis attack. As we mentioned above, the communication between agent and proxy server is very effectively guarded against sniffing attack or the possibility of correlation of requests and servers answers based on timing relations. In the communication between proxy server and destination Web server encryption is not a key. Eavesdropping of this data results only in the interception of information about the VAST proxy server's activity. If the system is employed by many users the information is worthless. In the extreme case, when there is only one VAST active user and when the eavesdropper has the possibility of interception of all requests realized by proxy server, the security **from others Internet users point of view** is the same as from the VAST service provider's point of view (still satisfied).

Let's consider the anonymity **from the VAST service provider's point of view** next. Referring to the figure of a sample communication between the VAST agent and the VAST proxy server (figure 2), attacker may only correlate particular requests such as it is shown in figure 4. We can observe that the proxy server may differentiate between particular sessions (three session noted in the picture with different shades). An attacker may not determine which shades represent user's activity. He also keeps in mind that user's requests correspond only to some blocks represented by the same (unknown) shade. Therefore, it is possible, after conducting an analysis of transactions content, to determine that data transmitted from sections: A, G, I, L, M is one session. B, D, F, J, M – 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 neither able to determine which one nor separate user's requests (in this example: B, F, M). Presented illustration is a schematic simplification. The regular blocks should not be identified with the volume of transmitted data. We should also consider the attack where VAST service provider resends fake pages or fake 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, but its result is the lost of reliability by service provider. After detection of this attack – what is easy and unavoidable – his 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.

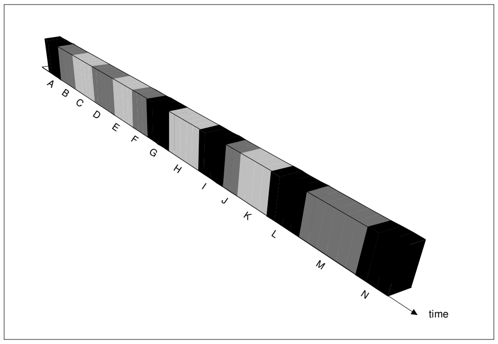

Fig. 4. Illustration of communication between agent and proxy server – from the proxy server's point of view (compare with fig. 2).

From the destination Web server point of view there is no ability to identify the system's users. To the destination Web server the only available data is about proxy server which forwards user's requests. Active elements placed on pages, which can communicate with destination server are removed by the VAST proxy server.

Until now no system providing effective protection against all of well known types of **traffic analysis attack** (like timing attack, message volume attack, flooding attack, linking attack) has been implemented. Systems based on David Chaum's theory, which are an accurate realization of MIXNET, provide an effective protection against timing attack or message volume attack. Below we are going to discuss the protection offered by the VAST system against each of the attacks. **Timing attack** is based on the observation of communication time through linking of potential end points and searching for correlations between beginning and/or ending of an event in each possible end point. The VAST system provides a total protection against this type of attack thanks to 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. Of course, in case where there is only one active system user (extreme case), eavesdropper can presume that all proxy requests come from the user. However, even in this case the anonymity of user is not compromised and it is the same as the anonymity of user from the VAST proxy server's point of view. **Message volume attack** is based on the observation of the transfer volume (i.e. message volume) and correlation of input and output. As mentioned above, the VAST system fills periods of user's inactivity with dummy traffic. Separation of particular messages from encrypted link between agent – proxy server is then practically impossible. **Flooding attack** is based on sending a large number of messages – flooding – or messages with certain characteristics by other system users. This is done in order to separate user's message. The VAST system protects against this attack because of the form of the message sent to the proxy server itself. Even after an effective isolation of user's message, it is still unknown which requests are generated by machine and which come from human. The VAST system, as presented above, does not offer an effective protection against **linking attack** which is based on a long-term observation. This attack uses changes in traffic related to presence of connection or its absence. In our VAST system concept, in order to maintain simplicity, we did not take into consideration these types of risks. However, it is possible to enhance the VAST system and include a mechanism providing effective protection against long-term linking attack (compare to section 6).

6. FUTURE WORK

The presented system introduces a concept which in its practical implementation requires additional mechanisms. When implementing this system for public use we should consider the possibility of an **long-term linking attack** (statistical profiling) conducted by proxy server or party which can eavesdrop on the communication of proxy. The risk originates in the possibility of separating recurring requests during the course of many sessions. To provide an effective protection against this class of attacks a mechanism of registering recurring requests should be used in the agent program. This would allow an introduction of dummy traffic simulating activity not only in the course of one session but during a longer time. It is important for the information gathered by the agent to be appropriately protected. The agent will record recurring user requests. The agent's program does not share this data with other parties. Recurring requests are accompanied by dummy traffic, which also simulates user's activity in the course of many sessions. The potential attacker will only be able to decide that the user is browsing through his favorite pages, but the attacker will not be able to determine through which one. We should consider the transformation of agent Java applet into a Java program which would be called local proxy. This will allow saving files locally on user's computer. It will be possible to save user's activity history. This will also permit storage of dummy traffic files and their usage as a cache memory. The user may choose page already downloaded during a dummy transaction. This greatly increases the navigation speed.

7. CONCLUSION

In this paper we have introduced an original method – VAST – which provides protection of Web user's privacy by granting versatile anonymity. This solution evolved from popular in WWW single proxy systems. It is a comprehensive technique which overcomes weaknesses of existing systems such as: serious, noticeable delays, access of service provider to user's private date and high costs of service implementation. The novel idea in this system – utilization of Web search engines resources to generate dummy traffic in the relation between local agent and distant proxy – may be in some cases viewed as its weakness. For users, whose fees for the Internet access are based on the amount of downloaded data, it means higher costs. We should stress that the system can block third party servers advertisement elements. It means that the graphic files from third parties are exchanged for dummy traffic. As usual – there is also a price for anonymity – to preserve full security, the user can not start navigating from direct URLs but from queries put into popular search engines. Another factor impacting the comfort of usage is that the requested phrases should be included in VAST's dictionary (which can be indeed very vast). It means that sometimes 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 VAST service provider's point of view is accomplished through masking. Therefore, the provider may with certain probability (chosen by the user) presume that particular requests come from user. We should note that total elimination of this weakness would mean the accomplishment of absolute WWW anonymity achieved by technical means, which seems to be practically impossible.

8. REFERENCES

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

[2] Chaum, D. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM Vol. 24 no 2,1981, pp. 84 - 88.

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

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

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

[6] Goldschlag, D. M., Reed, M. G., Syverson, P. F. Onion Routing for Anonymous and Private Internet Connections. Communications of the ACM Vol. 42 no 2, 1999, 39-41.

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

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

[9] Martin, D., Schulman, A. Deanonymizing Users of the SafeWeb Anonymizing Service. Privacy Foundation, Boston University, 2002.

[10] Reiter, M.K., Rubin, A.D. Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security, 1998, pp. 66-92

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