ANOSIP: Anonymizing the SIP Protocol

Iraklis Leontiadis
Institute Eurecom, Sophia Antipolis, France
leontiad@eurecom.fr

Constantinos Delakouridis
Athens University of Economics and Business, Greece
kodelak@aueb.gr

Leonidas Kazatzopoulos
Athens University of Economics and Business, Greece
lkazatzo@aueb.gr

Giannis F. Marias
Athens University of Economics and Business, Greece
marias@aueb.gr

ABSTRACT
Enhancing anonymity in the Session Initiation Protocol (SIP) is much more than sealing participants' identities. It requires methods to unlink the communication parties and relax their proximity identification. These requirements should be fulfilled under several prerequisites, such as time limitation for session establishment, involvement of several functional entities for session management, inter-domain communications and support of streaming services when the session is established. In this paper we propose the usage of a privacy enhancement framework, called Mist, as a solution to the anonymity issue in SIP. For achieving anonymity, the original Mist architecture was modified to be adapted in the SIP framework. We evaluate the adapted Mist framework to SIP and measure how efficiently it supports anonymity features.

Categories and Subject Descriptors C.2.2 **[Network Protocols]**: Applications; K.4.1 **[Public Policy Issues]**: Privacy

General Terms Design, Experimentation, Performance, Measurement, Security

Keywords Anonymity, Privacy, SIP.

1. INTRODUCTION
Every day new malicious actions on internet activity, available by exploiting the vulnerabilities of the end-systems or network protocols, are reported. The need to protect the personal freedom and privacy, achieve digital dignity, and, moreover, defend confidentially in the societal space, as well as in human relationships, is becoming more essential than ever. In this scope, privacy and anonymity over the Internet gained substantial consideration in the technical, procedural and legal domain. For every new service that is lunched and massively adopted in the Internet, privacy concerns arise immediately. The same applies for VoIP services, and especially for SIP which currently prevails in this new market. There are various reasons why an end-user wishes to maintain its anonymity when communicating using SIP. Firstly, a caller might wishes to conceal its identity at the receiver's phone. On the other hand, a callee might want to be unlinkable from her personal preferences and direct marketing campaigns. In its original specification, SIP supports anonymity, since the originator of a call could remain "Anonymous" to the callee, and for that reason default values are used when the user agent initiates a call. This feature supports caller anonymity against the callee, but not to the entire set of SIP realms, since practically the user agent server of the serving domain requires strong authentication of the caller. Additionally, using tunneling techniques, and especially end-to-end S/MIME encryption, selective anonymity can be supported. This option enables caller's privacy within the set of intermediate relays and the serving domains, if authentication is not required, but not against the callee. Finally, if network analysis tools are used in the network, then a malicious third party can track the locations, using the address-of-record fields, of the caller. In such a case it could link address-of-records to physical locations, using data mining techniques, and finally with people, since there would be only a few people that make phone calls from particular residential addresses during a day. So, the question is whether total anonymity is possible in SIP, and how this could be applied to shield the identity. In this paper we propose a new scheme for SIP protocol to enforce anonymity and privacy.

Our contributions. In this paper we adopt MIST anonymity architecture to SIP protocol. We evaluate our architecture under an attack scenario and calculate the anonymity based on the undermentioned architecture by obfuscating the identity and the location of the SIP client through tree architecture of MIST. We evaluate

Organization. The rest of the paper is organized as follows. Section 2 briefly describes the motivations of the paper. Section 3 reviews anonymity architectures. Section 4 presents the Anosip architecture. Sections 5 describes the attacking scenario whereby we measure the anonymity. Section 6 presents the results from the simulation of Anosip and section 7 concludes the paper.

2. MOTIVATION
To apply anonymity in SIP we should discriminate roles and actions. Even if various servers, intermediate proxies, and end-entities contribute on SIP, the set of actions, or service building blocks, that they contribute is actually

MPM'12, April 10, 2012, Bern, Switzerland.
ACM 978-1-4503-1163-2/12/04.

restricted. Subscription, registration, location (or redirection), call forwarding (or routing), call setup initiation-termination, and, optionally, authentication. This set of actions normally is performed by the entities belonging into two district sets of service providers: those of the callee and those of the caller. Thus, if we consider a model were an attacker wishes to reveal the identity of the calling parties, we can then define four legitimate parties in a SIP session: the caller, the callee, the service provider of the caller, and the service provider of the callee. In this direction we can define some privacy protection classes:

1. caller's absolute anonymity; the caller does not expose its identity to, or otherwise its identity cannot be exposed by, any other entity, or the attacker
2. caller's eponymity[1] only to the callee; the identity of the caller should be revealed only to the callee
3. caller's eponymity only to her provider; the identity of the caller should be revealed only to his/her provider
4. caller's eponymity only to callee's provider; same as above, but for the peer's provider

Except the first privacy class, the other three are not disjoint, and may coexist. In next sections we will see how the existing SIP anonymity proposal and specifications deal with these four classes. We should mention here that the potential attacker might be one of the service providers or the callee, depending on the privacy protection class. For instance, the attacker might be a callee that aims to expose the name of any caller who wishes not to display her name to the peer party. To support these privacy classes, any anonymity architecture should make an attacker unable to distinguish between the occasions when a callee transmits or receives a SIP message and the occasions when she doesn't. Additionally, it should take into account some characteristic of the SIP, such as:

1. the SIP messages should not be delayed
2. the sequence of SIP messages should not be violated
3. the traverse path of the SIP messages might be pre-determined, according to service agreements between local, regional and national operators

Moreover, any anonymity architecture should protect the physical location of the end-user. No one into system, neither the system itself, should know from which point a user is connected. Even if the relation of the transmitted or received SIP messages with a particular callee is not possible, the anonymity system should prevent attackers from linking the messages with physical locations. This will avoid the provable exposed conditions, whereas an attacker can prove the identity of the sender to others. For instance consider a user who decided to use anonymous SIP features. The UAC uses a meaningless URI, such as sip:thisis@anonymous.invalid [5]. If this meaningless URI is always used for this particular user, then it is possible to intercept SIP traffic, and connect this URI with different "Addresses-of-Record" (AoR). Then, using commercial or open source tools the attacker will link these AoRs with physical locations, and then with end-users' identities.

[1] This is a Greek word, actually an antonym of anonymity

3. ANONYMITY ARCHITECTURES

To enhance or provide privacy in the internet services several privacy enhancement technologies (PET) have been proposed. Chaum's Mixes [1], Stop-and-Go Mixes and MixeNets [2], Crowds [3], Hordes [4], Onion Routing [5], and Mist [6] are some of the preserving techniques. Tor [13] has gained the interest of many researchers as it is becoming a standard architectur for anonymous web browsing. The idea behind Tor is based in Onion routing. A user selects a number of realaying nodes which encypt and send the user data to the final destination obfuscating the end-to-end path of the transmission.

For the most of these PET approaches, applied mainly for e-mail and asynchronous web communications, there are some deployment difficulties when adapted to SIP. Latency is an issue, since SIP a call setup request, e.g., an INVITE, requires immediate response. This feature is not supported directly. Additionally, these PETs do not support bidirectional communications, excluding the onion routing, a characteristic that is essential for SIP. Moreover, anonymity should be semantically supported. In that sense, the PET mechanism should support unlinkability of location where calls are initiated (or terminated) from SIP URIs, or physical addresses (e.g., IP addresses). The most of the previously mentioned PETs support anonymity in transit, and do not have means to support unlink-ability.

A promising privacy system that overcomes these drawbacks is the Mist. The Mist [10] handles the problem of routing a message through a network while keeping the sender's location private from intermediate routers, the receiver and potential eavesdroppers. The system consists of a number of routers, called Mist routers, ordered in a hierarchical structure. According to Mist, special routers, called "Portals", are aware of the user's location, without knowing the corresponding identity, whilst "Lighthouse" routers are aware of the user's identity without knowing her exact location.The "Lighthouse" routers, hereafter will be referenced to as LIG.

4. ANOSIP ARCHITECTURE

SIP protocol specification suggests that the Home Server (Registrar, Redirect, or Proxy server) keeps knowledge of both user's ID and current location. Our goal is to distribute this knowledge to more than one entity. If though, it will be difficult for eavesdroppers to inference user's location information. Since a SIP user registers to Home Server (using her ID) and this server is the one that all SIP entities refer to in order to locate the registered user, we could consider that Home server corresponds to user's Mist LIG. Furthermore, we define as Mist Portals all the Remote SIP servers that user is connected to in order to establish communication through SIP. In general, we presume that each SIP server (hereafter called MSIP Server) can act as Mist LIG (for the users that have been registered to it), Portal (for the users that at some point can connect to the SIP network) or Mist router. To enforce Mist to support anonymity in SIP, small modifications are required in SIP. Currently, SIP location service is an LDAP directory that keeps the current physical position of registered users. However by applying Mist, the location of the users is no

longer known to Home Server. Instead, the latter will have knowledge of a way to route packets to the user. In terms of Mist, the Mist user's binding table can be used to replace the location service. This table keeps routing information about the Mist communication circuits with each user. Furthermore, we consider that:

1. A Mist Hierarchy has been applied. Mist Hierarchy considers that all Mist servers are ordered in a tree-based hierarchical structure. However, to apply Mist routing in SIP, we have to alter this structure by adding connections between the siblings of each level of the tree. Thus, a MSIP server is able to forward packets apart from its ancestor, to its siblings. The reason for this modification is discussed later.

2. A PKI has been established, pairs of keys have been created, and the corresponding public-key certificates have been distributed to MSIP servers. Furthermore the authentic public keys are accessible from every MSIP Server. Additionally, each user holds a pair of keys, related only to the user's nick name (using e.g., anonymous certificates) and not real-life information.

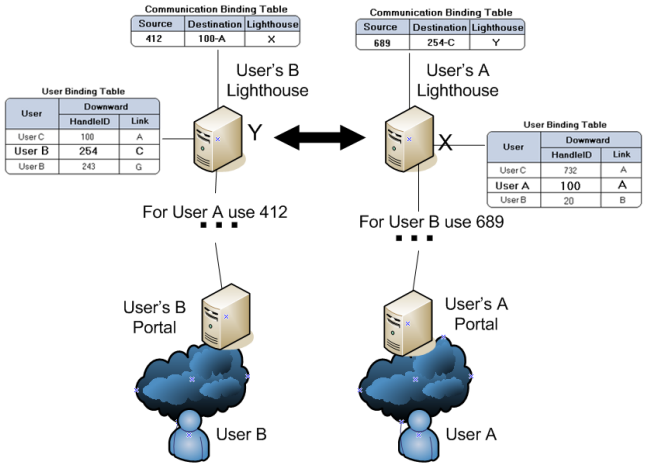

Figure 1. Establishment of a Mist circuit

4.1 Registration procedure

Suppose Alice has a WiFi connected laptop and place calls using SIP. When she triggers this service for the first time, her handset initializes SIP Registration routine and connects to the first available Registrar SIP server. During this registration phase Alice is prompted for personal information, nickname and password. Note that since she doesn't wish to reveal her real identity, she registers to the system hiding her personal information. However, she will use the nick name "Mother", so that her friends that know her nick name can call her. From a Mist point of view, the registrar SIP server considers to be her LIG. The LIG will be the point of contact for other SIP users in order to get in touch with her. Upon registering, the LIG sends a Mist notification to the Lookup Service to inform it that user "Mother" has been registered to this LIG.

4.2 Mist circuit establishment

Alice is visiting a friend on the other part of the city and wants to be reachable by SIP clients but not traceable. She connects to the first available SIP server. From the Mist point of view, this considers to be her Portal. Next step is to setup a Mist Circuit between Alice Portal and LIG. Note that the Portal, contrarily to the original Mist procedure, does not forward to Alice's laptop the list with all the available LIGs since, as we mentioned earlier, Alice's LIG is the SIP server where she was originally registered to (i.e. the home registrar server of the SIP protocol). Accordingly, her laptop encrypts a predefined message with the public key of the LIG and forwards it to the Portal. The latter routes this update packet to her LIG. Note that since Alice LIG is predefined, it is likely that this LIG is not an ancestor of her Portal. To ensure that the update packet will reach the LIG, regardless its position on the tree, the Mist Portals forward packets to their ancestors, as well as to their direct connected siblings. In more details, if the MSIP server receives a packet from its predecessor, it forwards the packet to the ancestor and to the directly connected siblings. Otherwise, if it receives a packet from a sibling server it forwards the packet to the next sibling. Upon receiving the update packet, the LIG stores the Mist circuit information to the user binding table. At this point, the Mist circuit has been established. The LIG is able to forward packets to Alice (actually to "Mother") without knowing her exact location.

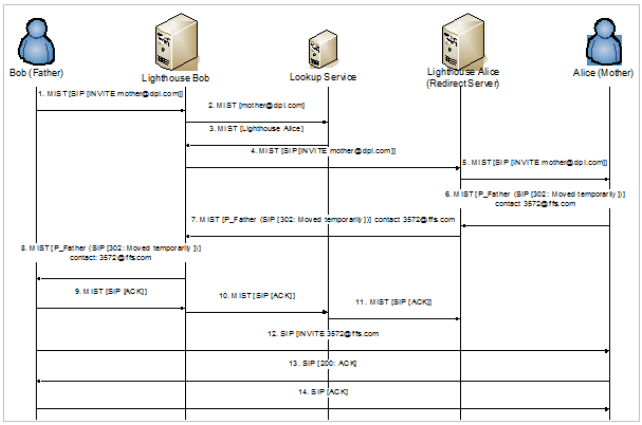

Figure 2. Establishment of a Mist circuit

4.3 Sip session

Suppose that Bob wants to call Alice. Both users have established a Mist circuit with their corresponding LIGs. Bob is aware that Alice's nickname is "Mother". The call establishment procedure is as follows:

1. Bob (who has registered with the nick name "Father") creates a MIST Packet towards his SIP LIG and encapsulates a SIP INVITE request for the user "Mother". He sends this up to the Mist Hierarchy.

2. Bob's LIG receives the packet, determines the destination user and searches the Lookup Service for the corresponding LIG.

3. Bob's LIG creates a Mist Packet towards the Alice's LIG and encapsulates the INVITE that he received.

4. Alice's LIG receives the packet, which is a SIP Redirect Server, determines that the called person is "Mother" and looks in the binding table to locate her.

5. Upon retrieving the Mist routing information, it creates a Mist packet with the SIP INVITE request and sends it to her through the Mist circuit.

6. Alice receives the packet, determines that it is an INVITE request from her friend Bob (she knows that his nickname is "Father").

7. Alice creates a SIP Redirect Packet to inform Bob about her current location, encrypts this message with Father's public key, and encapsulates everything in a Mist Packet towards her LIG. The public key of Bob is based on his nickname to enforce his anonymity.

8. Alice's LIG upon receiving the packet, it determines that the destination is Bob's LIG, encapsulates the content of Alice's packet to a Mist Packet and send it to Bob's LIG

9. Bob's LIG forwards the packet to Bob.

10. Bob, upon receiving, creates a SIP packet to ack. At this point, Bob knows Alice remote current address

11. Therefore the next step is to send directly to her an SIP INVITE request.

They both acknowledge, the SIP circuit is formed, and they have an established call.

Taking in account the untraceability of the packets routed through the Mist and the distribution of knowledge (i.e., Portals know "where", LIGs know "who") we can preserve the privacy of the location of the users. Furthermore, considering only users that are registered to the system using their nickname, and realistically assuming that the corresponding private keys have been issued based on this nickname, anonymous communications are actually supported.

The aforementioned technique assumes that the communicating parties have knowledge of each other (Bob knows that Alice's nickname is "Mother" and vice versa).A problem arises when the two parties have no prior knowledge. Therefore, we introduce the Trusted Third Party directory which is called "ID Directory". This directory stores tuples with the following format:

<NickName A, ID A>

Where ID A is the user's real name e.g. "Alice" and NickName A is the nickname that the latter is using e.g "Mother". Should someone i.e. Alan wants to communicate with Alice for the first time, he presents to the ID Directory a authentication credentials in order to use the requested tuple. After validating user's credentials it will send to Alan the corresponding tuple for Alice. During ANOSIP communication establishment, Alice receives on step 6, an invitation from "Grandfather" which is Alan's nickname. Alice inquires the ID Directory for the tuple <"Grandfather","Alan"> using the aforementioned authentication procedure. Upon receiving the corresponding tuple, she decides whether or not to answer back to Alan.

5. Attacking scenario

Shannon [11] introduced entropy as an information theoretic concept that provides a measure of the uncertainty of a random variable . Let X be a discrete random variable with probability mass function $p_i = Pr(X = i)$, where i represents any possible value that X may take with probability $p_i > 0$. We denote by $H(X)$ the entropy of a random variable, and by N the number of subjects in the anonymity set (i.e., Lighthouses, Portals and Routers). $H(X)$ can be calculated as:

$$H(X) = \sum p_i * \log(p_i)$$

In our case, p_i is the probability of SIP user i being linked to an identity and location (since knowledge of only the location or only the identity of the user is supported in MIST). Thus, our analysis is focused on the uncertainty of connecting a user associated with a particular attribute (i.e. identity) to a particular place (e.g. IP address). So an effective attack scenario is the one that some of the nodes m in the number in the network collude to associate identity and location. Obviously, $N=P+R$, where P and R is the number of Portals, and Routers in the Mist hierarchy respectively. A user u_i is served by one Portal (P_i) and R_{ir} Routers (vector $R_{r1}, R_{r2}, \ldots, R_{rk}$), $R \geq r \geq 1$. The p_i probability is defined as:

$$p_i = \frac{N - R - \frac{m}{P} * pf}{N}$$

where pf is the probability of a node to be controlled by an intruder.

This entropy definition was enhanced in [10] and the effective anonymity concept was introduced. When launching an attack against an information system, the attacker's goal is to evaluate, with high precision, the distribution of probabilities that link any distinguished subjects (e.g., user, process, transaction) to the particular item of interest (target). According to [11] different subjects might illustrate higher or lower probability p_i to link with the target, depending on the information obtained by the adversary using the attack, or actual relationships. So, if N is the total number of subjects which are linked by the adversary to the target with a non-zero probability (i.e., $pi > 0, i = 1..N$) then the effective anonymity set size is defined as the entropy $H(X)$ of the distribution X of probabilities that link the subjects of the anonymity set to the target. In [10], the entropy is normalized to express a degree of anonymity in the scale $0, \ldots, 1$. The effective anonymity set size is maximized if all the N subjects are connected with equal probability (i.e., $p_i=1/N$) to the target. This corresponds to the maximum entropy, denoted by H_{Max} where $H_{Max}=log_2(N)$. The amount of information gained by the adversary with an attack is the difference in the entropy before and after

the attack, that is: $H_{Max} - H(X)$. The degree of anonymity d is defined as the normalized value of this difference:

$$d = 1 - \frac{H_M - H(x)}{H_M} = \frac{H(X)}{H_M}$$

Figure 3 depicts the anonymity degree as previously described. We randomly choose the value of the *pf* probability. It is obvious that the more the colluding portals along with the higher *pf* the less the anonymity degree. As such there is a trade-off between the number of the intermediate routers and the degree of anonymity.

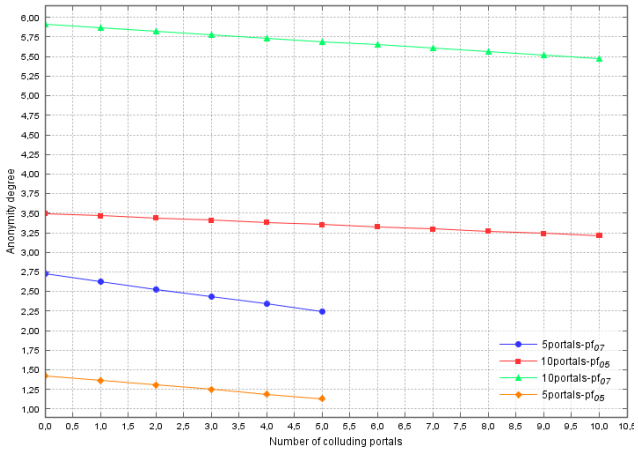

Figure 3. Anonymity degree of ANOSIP

6. SIMULATION RESULTS

Once the mist circuit is created between the lighthouse and the user then the lighthouse operates as a representative of the user. This correlation between the Lighthouse and a user should be saved in each domain to enable questions from other Lighthouses to be answered from Lighthouse located in the domain of each user. To implement the location service we have used the database MySql 5.0.67. The Mist entities that communicate with the location service are the Lighthouses. Using the "Hibernate" library we have created an interface for the lighthouse to record and query the location service. The relational table used in the form <Lighthouse, User>. The "hibernate" library aims to link the objects created in an object-oriented programming language (Java) to the tables of a relational database.

For the efficiency of the architecture we have used a traffic generator for SIP messages, the SIPP v 3.1 [8]. In this scenario (Figure 4) we have assumed the following category of SIP chats: Children equipped with mobile phones are communicating with their parents or their supervisors to indicate their location for safety reasons. This is a real user scenario in schools trips. To reduce the cost of communication the children are connected in a wireless-access point and through SIP clients they talk with their supervisors. In such situations 30secs of conversation are sufficient to indicate the real location of the children or to mention an emergency situation as fire or earthquake. The SIP traffic generator sends registration requests to SIP registrar server and we are interested to see our response times for successful listings. The scenario run for 30 seconds at a rate of 10 calls per seconds.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<scenario name="register_client">
<send retrans="500" start_rtd="1" start_rtd="2" start_rtd="3"
start_rtd="4" counter="1">
<![CDATA[ REGISTER sip:[remote_ip] SIP/2.0
Via:SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From:<sip:[field0]@[field1]>;tag=[call_number] To:
<sip:[field0]@[field1]> Call-ID: [call_id] CSeq: 1 REGISTER
Contact: sip:[field0]@[local_ip]:[local_port] Max-Forwards: 5
Expires: 1800 User-Agent: SIPp/Linux Content-Length: 0
]]>
</send>
<recv response="500" rtd="1" optional="true" counter="2"> </recv>
<recv response="503" rtd="2" optional="true" counter="3"></recv>
<recv response="504" rtd="3" optional="true" counter="4"></recv>
<recv response="200" rtd="4" counter="5"> </recv>
```

Figure 4. The SIPp scenario

In Figure 5 we can see the response times for successful registrations for the all the registration requests. We note that the first responses come quickly and then they are slow enough (2.5-3 secs). At the vertical axis there is the time in ms and at the horizontal is the time period for 30seconds that the experiment was running.

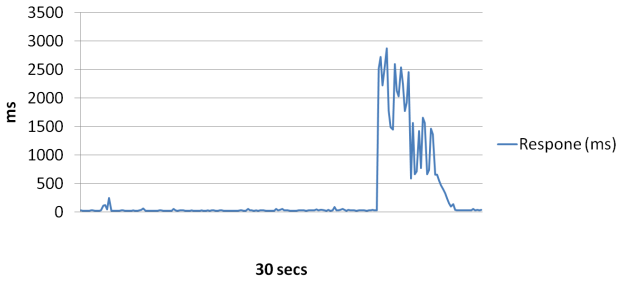

Figure 5. Registration response times

As the following table illustrates, the majority of successful responses on SIP Invite (i.e., OK method) are less than 40ms. That indicates an negligible extra overhead to the registration precedure through MIST.

Table 1. Response Time on SIP messages with MIST

Response of 200 calls	#number of responses
0ms-10ms	0
10ms-20ms	0
20ms-30ms	58
30ms-40ms	39

7. SUMMARY AND FUTURE DIRECTIONS

In this paper we have presented ANOSIP, a privacy enhancement framework for the SIP protocol. We have taken into account a specific privacy threat: the correlation of a session or a receiver with a specific caller. We have adopted in ANOSIP the Mist privacy framework, proposing significant enhancements and modifications to overcome

weaknesses and strength the anonymity of the Mist users. We have analyzed ANOSIP by measuring the degree of anonymity based on different attacking scenarios. Finally, some preliminary simulation results show that Mist does not introduce significant delay overhead during operation in special use case scenarios whereby the caller seeks to learn the location of the callee in emergency cases as fires, earthquakes, children awareness by their supervisors or parents, in which the communication overall time is around 30seconds. Our future directions include the simulation of more complex calls with a SIP traffic generator and the deployment of ANOSIP to planetlab [12] as a VOIP infrastructure.

8. ACKNOWLEDGMENTS

Most of the work has been done while Iraklis Leontiadis was at Athens University of Business and Economics in Mobile Multimedia laboratory. The authors thank the comments of the anonymoys referees.

9. REFERENCES

[1] D. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms”, Communications of the ACM, Vol. 4, No. 2, Feb. 1981

[2] D. Kesdogan, et al, “Stop-and-go MIXes Providing Probabilistic Security in an Open System”, 2nd Intl. Workshop on Inform. Hiding, 1998

[3] M. K. Reiter and A. D. Rubin, “Crowds: anonymity for web transactions”, ACM Trans. Inform. Systems Security, 1(1):66.92, 1998

[4] B.N Levine and C. Shields, “Hordes: A multicast-based protocol for anonymity”, J. of Computer Sec., 10(3):213-- 240, 2002

[5] M.G. Reed, P.F. Syverson, and D.M. Goldschlag, “Anonymous connections and onion routing”, IEEE JSAC, Vol. 16, Issue: 4, 1998

[6] J. Al-Muhtadi, et al., "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments", in Proc. Intl. Conf. of Distributed Comp. Syst., 2002

[7] A. Serjantov, and G. Danezis, Towards an Information Theoretic Metric for Anonymity, Dingledine and Syverson (Eds.), Designing Privacy Enhancing Technologies, LNCS 2482, pp. 41-53, 2002 - the This entropy definition was enhanced in

[8] SIPp. http://sipp.sourceforge.net/

[9] C. Dıaz “Anonymity Metrics Revisited”, Dagstuhl Seminar on Anonymous Communication and its Applications, October 2005

[10] C. Dıaz, and S. Seys, and J. Claessens, and B. Preneel, Towards measuring anonymity, Dingledine and Syverson (Eds.), Designing Privacy EnhancingTechnologies, LNCS 2482, pp. 54-68, 2002

[11] C. E. Shannon, A Mathematical Theory of Communication, The Bell System Technical Journal, volume 27:379–423, pp. 623–656, 1948.

[12] PlanetLab. http://www.planet-lab.org/

[13] Tor project, http://www.torproject.org