Providing Anonymity Services in SIP

L. Kazatzopoulos, C. Delakouridis, G.F. Marias
Dept. Of Informatics
Athens University of Economics and Business
Athens, Greece
lkazatzo@aueb.gr, kodelak@aueb.gr, marias@aueb.gr

***Abstract*—Anonymity in telecommunication services means much more than protecting the identity of participants. It requires mechanisms and protocols that unlink the communication parties, unlink users from their location, and avoid statistical analysis. These functional requirements apply also when providing anonymity services to SIP, whereas the identities of caller and the callee(s) should be secured. On the other hand, SIP introduces additional functional requirements to any anonymity services, such as time limitations for session establishment, involvement of several functional entities, inter-domain communications and support of streaming services when the call is established. Here, 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. The paper discusses how Mist can be adapted to SIP and efficiently support anonymity features.**

Keywords: VoIP, SIP, privacy, Anonymit, Mist

I. INTRODUCTION

In nowadays Internet, end-users should employ technical and procedural means to defend against attackers that maliciously survey, or spy, the Internet using network traffic analysis tools. This will protect the personal freedom and privacy, achieving digital dignity, and, moreover, defend confidentially in business, as well as in human relationships. In this scope, privacy and anonymity when communicating 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 issues 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 for displaying in the receivers' hone, a feature that is usually used in mobile phone calls. On the other hand, a callee might want to be un-linkable from her/his 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 tunnelling 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 parry 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 whereas total anonymity is possible in SIP, and how this could be applied to shield the identity, or the character of a dog or a human.

II. MOTIVATION

According to Justice L. Brandeis, the "right to privacy" is "the right to be left alone". Alan Westin identifies privacy as "the desire of people to choose freely under what circumstances and to what extent they will expose them-selves, their attitude and their behavior to others". Nowadays, we can define privacy in different domains, not vertical, but overlapped:

- physical privacy – such as DNA searching
- information privacy – the unsanctioned invasion of privacy by the government, corporations or individuals in order to identify, or even handle, our personal information such as our age, address, market profiles, daily communications, or even sexual preference.
- context privacy – each individual's fundamental right not to be unlinked with places, people, locations and preferences in his daily live because of surveillance cameras, sensor networks and RFID systems.

Privacy is sometimes related to anonymity. According to [1] and the Oxford Dictionary, anonymity is defined as .the state of being anonymous which in turn is described as .nameless, having no name; or unknown name. This definition arises some vagueness, since in real world implementations it should be clear which identity should be hidden and from whom. To make the scope of anonymity more undoubtedly, Pfitzmann and Kohntopp introduced the most common definition of anonymity used in the information and information community [2]: "anonymity is the state of being not identifiable within a set of subjects, the anonymity set". The anonymity set is a sensible metric since it associates the sender or receiver anonymity with a set of people and their actions. For instance the receiver anonymity set

is the set of people who could have received a message intercepted by an attacker. Obviously the cardinality of the anonymity set is a measure of anonymity. A user is .k-anonymous, or has k-anonymity, if he/she is one of at least k users within a specific anonymity set associated with a particular action. Recently, Serjantov and Danezis defined an information theoretic measure of anonymity [3]: each member of the anonymity set is assigned a probability equal to the likelihood that the member performed the anonymous action.

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 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:

- 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
- caller's eponymity[1] only to the callee; the identity of the caller should be revealed only to the callee
- caller's eponymity only to her/his provider; the identity of the caller should be revealed only to his/her provider
- 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 that wish not to display his/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/he do not [4]. Additionally, it should take into account some of the idiosyncrasies of the SIP, such as:

- the SIP messages should not be delayed
- the sequence of SIP messages should not be violated
- the traverse path of the SIP messages might be predetermined, 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 [4], 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.

III. ANONYMITY ARCHITECTURES

To enhance or provide privacy in the internet services several privacy enhancement technologies (PET) have been proposed. Chaum's Mixes [6], Stop-and-Go Mixes and MixeNets [7], Crowds [4], Hordes [8], Onion Routing [9], and Mist [10] are some of the preserving techniques.

Mixes [6] introduced the notion of anonymous digital communication. The Mix system provides unlinkability of sender and receiver. This ensures that while an attacker is able to determine that the sender and receiver are actually sends or receives messages, she/he cannot determine whom they are communicating with. The system consists of special mix nodes which store, mix, and then forward the messages in transit. The sender predetermines the route of the message through one or more mix nodes using a well-defined protocol. A public key cryptography protocol is also used to ensure that any message cannot be tracked by an attacker as it passes through the mix network. In their simplest form (called a threshold mix) a mix node waits until it collects a number of messages as input. It uses its private key to reveal the address of the next mix node (or final destination) and reorders the received and buffered messages by some metric before for-warding them. In that sense, omnipresent attacker cannot trace a message from its source to its destination without the collusion of the mix nodes. To provide a mix-network routing protocol, Kesdokan et al. introduced the Free Route and Mix Cascade concepts [7]. The former gives autonomy to the sender for dynamically choosing the trust path of the mix-nodes, whilst in the latter the routing paths are predefined. Mix networks introduce delays due to buffering and mixing and different padding patterns for mixing real with dummy traffic. Continuous Mixes try to avoid the delay issue by introducing fixed delay distributions for buffering and mixing. Mixes became subject to several attacks, such as timing at-tacks [11], statistical analysis of message distribution [12], statistical properties of randomly constructed routes [13] [14], and packet flow correlation attacks [15] [16].

Crowds [4] is a network that consists of voluntarily collaborating nodes. It is based on the idea that our anonymity can be protected better when we moved ourselves within a crowd. According to [4], Crowds' web servers are unable to learn the true source of a request because it is equally likely to have originated from any member of the crowd. Even collaborating crowd members cannot distinguish the originator of a request from a member who is merely forwarding the request on be-half of another. In Crowds each user (browser) is represented in the

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

system by a jondo process. Each message that needs anonymity enters into the Crowd node, its presence is announced via the local jondo, and is sent to another, randomly chosen, jondo with probability p, or to the actual server with probability 1-p. When the server (recipient jondo) receives the message it answers using the same, forward, path. Crowds can face effectively trace back attacks, and it can mitigate collusion attacks if the users select randomly the set of forwarding jondos.

Onion Routing [9] is an overlay infrastructure for providing anonymous communication over a public network. It supports anonymous connections through three phases: connection setup, data exchange, and connection termination. In the setup phase the initiator creates a layered data structure, called on-ion, which implicitly defines the route path. An onion is recursively encrypted message using public key crypto. The number of encryptions is equal to the number of the onion routers that this structure passes towards the destination. The outer cryptographic control information refers to the first onion router in the path, whilst the inner cryptographic control information refers to the last onion router in the path. Each onion router along the route uses its public key to decrypt the onion that it receives. This operation exposes the embedded onion, and as a result, the identity of the next router. Once the onion reaches the destination, all the inner control data are appeared as plaintext. This establishes the anonymous end-to-end connection, and then data can be sent in both directions. As data are routed through the anonymous end-to-end connection, each onion-router removes one layer of encryption, so the data arrives as plaintext at the recipient. This layering occurs in the reverse order (using different algorithms and keys) for data moving backward. All the messages illustrate identical sizes and arrive at an onion router at fixed time intervals. They are mixed to avoid correlation by potential attackers. Additionally cover traffic deludes external eavesdroppers. Onion Routing effectively resists traffic analysis.

Hordes [8] is an anonymity infrastructure that combines elements os Onion Routing and Crowds. It is the first protocol that uses multicast transmission, when the destination answers to the sender. It includes two phases, the initialization and the transmission phase. In the first phase, Hordes borrows the jondos idea from Crowds, and a public key scheme is used to add authentication services. The sender also sends a join request message to a proxy server. The proxy authenticates the sender, it returns a signed message that contains the multicast address of jondos and informs the multicast group for the new entry. In the second phase, for the transmission of a message the sender selects a subset of jondos for the forwarding path and a multicast group address for the backward path. When data message is scheduled for transmission, the sender chooses a jondo member of the forwarding subset and sends this message as an encrypted data structure. The chosen jondo sends this message to another, randomly chosen, jondo with probability p, or to the receiver with probability 1-p, using encryption layers as well. The receiver replies on the backward path; it sends an acknowledgment as plaintext message to the multicast group.

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 unlinkability.

A promising privacy system that overcomes these drawbacks is the Mist. The Mist [10] handles the problem of routing a message though 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 his/her exact location. We will discuss Mist in more detail in the next section.

When practical issues arise, proxy servers offer anonymity services on the World Wide Web. For instance the Anonymizer.com provides proxy services via rewriting URLs such that a link to, e.g., http://www.you.gr might be rewritten as https://anonymity.proxy.net/www.you.gr. SSL encryption is used to ensure confidentiality of the connections between the end-user and the proxy server.

On the other hand, a theoretical model for ensuring anonymity is the k-Anonymity concept. [17], [18] which was originally introduced in the context of relational data privacy. It ad-dresses the question of "how a data holder can release its private data with guarantees that the individual subjects of the data cannot be identified whereas the data remain practically useful" [19]. Regarding LBSs and mobile clients, location k-anonymity refers to the k-anonymity usage of location information. A subject's location is considered k-anonymous if and only if the location information sent from a mobile client to LBS is indistinguishable from the location information of at least k -1 other mobile clients [20]. The location perturbation is an effective technique for supporting location k-anonymity and dealing with location privacy breaches exemplified by the location inference attack scenarios. If the location information sent by each mobile client is perturbed by replacing the position of the mobile client with a coarser grained spatial range such that there are k-1 other mobile clients within that range k>1, then the adversary will have uncertainty in matching the mobile client to a known location-identity association or an external observation of the location-identity binding. This uncertainty increases with the increasing value of k, providing a higher degree of privacy for mobile clients.

IV. Existing Proposals for Anonymity in SIP

In the original specification of SIP the anonymity feature was juvenilely supported. To enable anonymity, the UAC should use in the "From" header field the display name "Anonymous", along with a syntactically correct, but otherwise meaningless URI (e.g., sip:an@anonymous.user). Additionally, tunneling encryption is suggested for anonymity. This is achieved by encrypting the header fields, and producing an

outer, new, “From” header filed that includes the "Anonymous" value in the display name subfield. This end-to-end encryption is not immune to location tracing attacks, since statistical analysis of sniffed data might reveal the communicating parties. A more recent draft RFC introduces guidelines for the creation of messages that do not reveal personal identity information, and a new "privacy service" logical role for intermediaries is defined to answer some privacy requirements [21]. Additionally, an internet draft proposal suggested the extension of the SIP that enables parties in a SIP session to be identified by different types of party information, which are authenticated by a trusted entity [22]. These trusted entities are dele-gated by end-users to reveal the identity of the calling or called party to peer entities. Trusted peers might receive in-formation that identifies an end-user, if these entities are sup-posed to provide the same level of privacy, i.e., to reveal party information to other trusted peers. Sipanon is another SIP anonymity proposal that introduces an architecture two user agents (a User Agent Client and User Agent Server) that are coupled back to back (B2BUA) [23]. A message from the anonymous user's UAC is received by an Anonymizer's UAS. This message is then anonymized. The “From” header is changed and sent out from the Anonymizer's UAS to the remote UAC. The Anonymizers maintain a store of mappings between “real" SIP addresses and “anonymous".

V. A New Proposal for SIP Anonymity

A. *Mist at a glance*

The key point of the Mist architecture is the distribution of knowledge. The “Lighthouse” routers, hereafter referenced to as LIG, are aware of the user’s identity without knowing the exact location. “Portal” Routers are aware of the user’s location, without knowing the corresponding identity. Furthermore, due to the decentralized Mist architecture, a possible collusion between the aforementioned Mist routers is extremely difficult since the routers are unaware of each other’s identity. The architecture is applicable to a variety of network facilities since it uses a general purpose protocol that enables privacy on IP networks despite the underlying technology (e.g., Ethernet, 2/3G Mobile). In short, the Mist architecture consists of a number of routers, called “Mist routers” ordered in a hierarchical structure. A typical structure is shown in Fig. 1. The leaf nodes in the hierarchy are called Portals which act as connection points where users can connect to the Mist system. Let us assume that user A requires a network connection that ensures privacy and data confidentiality. User A has to register himself to the Mist System. His device locates and interfaces directly with one of the available portals in the surrounding physical space. The Portal, upon receiving a registration request, replies with a list of its ancestral Mist Routers that exist at a higher level within the Mist hierarchy and are willing to act as a LIG for the user. A LIG is a Mist Router that acts as a point of contact for user A. Users that intend to communicate with user A have to contact his LIG.

Following LIG selection, a virtual circuit (Mist Circuit) is established between user A and the corresponding LIG. This process, called “Mist Circuit Establishment”. It aims to entitle user’s A LIG to authenticate A without revealing his/her physical location. At the same time, it hides from Portal, the user’s identity and the designated LIG. Furthermore, the Mist Circuit applies a hop to hop handle-based routing technique for packet transmission between source and destination nodes and in combination with data encryption manages to conceal from the intermediary nodes information related to the identities and location of the communicating parties. To establish a Mist Circuit, user A generates a “Mist Circuit Establishment” packet and transmits it to the corresponding Portal, without informing the portal of the selected LIG. The Portal, upon receiving the packet, assigns a special number, called handle ID, to the communication session with user A.

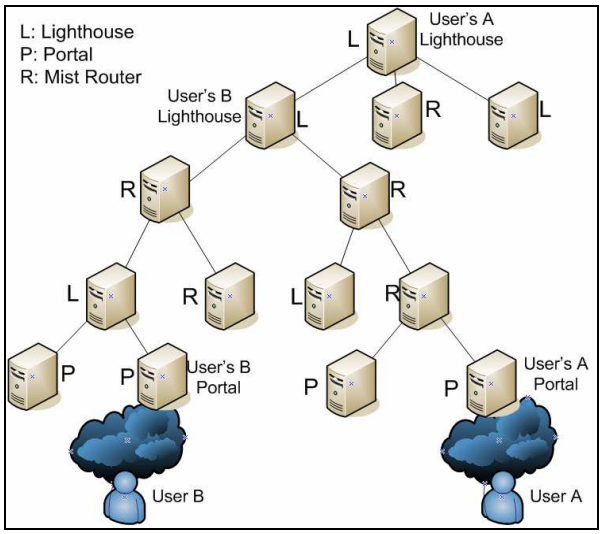

Figure 1. Typical Mist structure

Thereafter, the Portal encloses the assigned Handle ID to the received packet and forwards it to its Mist Router ancestor. As the packet propagates through the Mist hierarchy each LIG Router, attempts to decrypt the payload using his private key. If the decryption fails, the particular router infers that it is not the recipient of this packet and thus, forwards it to the next router on the hierarchy. This process is repeated on each of the intermediate Mist Routers until the packet reaches its final destination. In case the decryption of the payload is successful, this indicates that the user selected the current Mist Router to act as his LIG. Finally, the LIG answers back to user A and confirms the registration. From this point, a secure circuit is established through which user A can communicate securely with his LIG. Note that even though the LIG of user A can infer that his/her physical location is underneath Mist Router “Y”, it is very difficult to determine the exact position. Following circuit establishment, the LIG undertakes the role of representing the end user. An issue that has to be addressed is the detection of the user’s LIG. A public directory (i.e. LDAP server) or a WEB server can be used for that purpose.

Assume now that user B intends to communicate with user A (Fig. 3) and both of them have previously established a Mist circuit with LIG B and LIG A, respectively. User B transmits to LIG B a packet indicating that he/she wants to set up a connection with user A. LIG B verifies that the originator of the message is B, locates the LIG of user A and performs the initialization procedure for connection establishment with the LIG B. As soon as the communication path is established, users A and B are able to communicate. The intermediate routers are unaware

of the two ends of the communication. Moreover, it is impossible for B to determine the location of A, and vice versa.

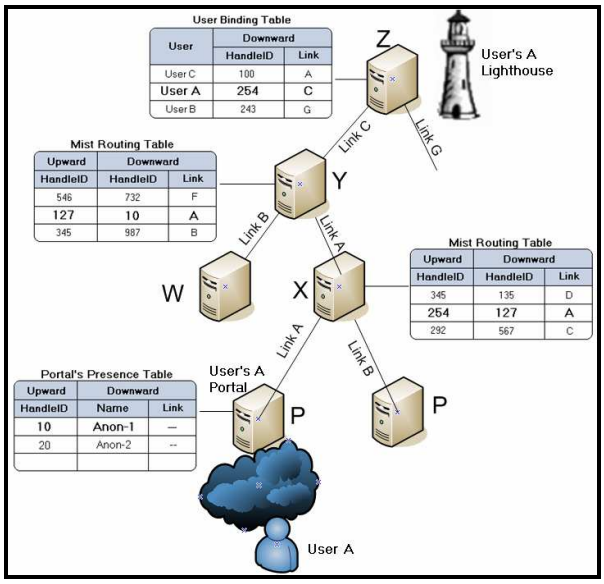

Figure 2. The Mist Circuit Establishment

B. *Applying MIST in SIP*

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 registered to it), Portal (for the users that at some point can connect to the SIP network) or Mist router. To apply 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 not 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.

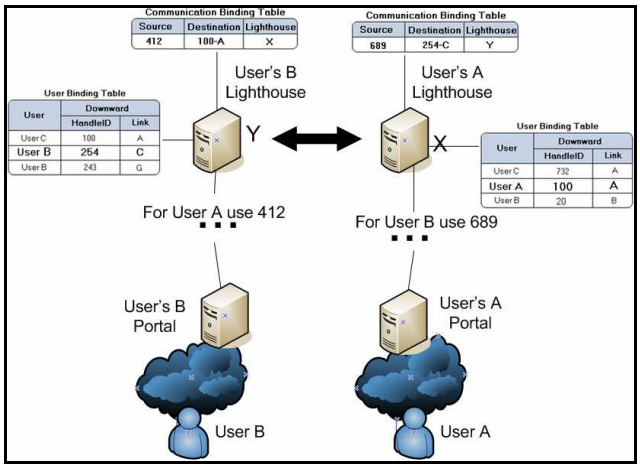

Figure 3. Establishment of a Mist circuit

C. *Registration Phase*

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.

D. *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.

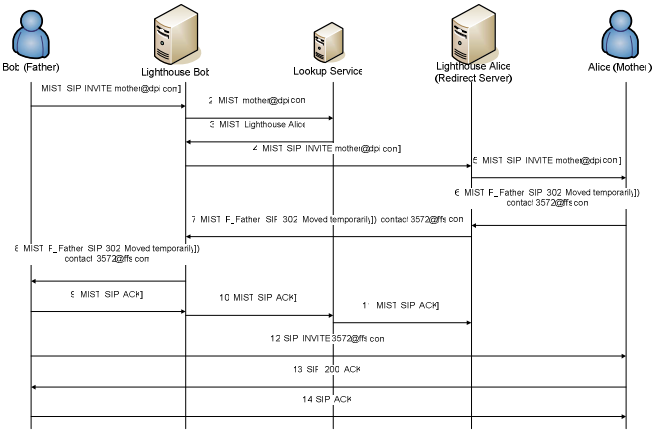

Figure 4. Applying Mist in SIP

E. Establishing a 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.

12) 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.

REFERENCES

[1] S. Rana and J. Sharma, "Frontiers of Geographic Information Technology", Springer Berlin Heidelberg Publishers, 2006

[2] A. Pfitzmann and M. Khntopp, "Anonymity, unobservability, and pseudonymity: A proposal for terminology", draft v0.21, Sept. 2004

[3] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity", in Privacy Enhancing Technologies, LNCS, Apr. 2002

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

[5] RFC 3261, J. Rosenberg, H. Schulzrinne, G. Camarillo et al., "SIP: Session Initiation Protocol", IETF June 2002

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

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

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

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

[10] 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

[11] L. Øverlier and P. Syverson, "Locating hidden servers" In Proc. IEEE Symposium on Security and Privacy, 2006

[12] G. Danezis, "Statistical disclosure attacks", In Proc. Security and Privacy in the Age of Uncertainty, volume 250 of IFIP Conf. Proc., 2003

[13] M. Wright, M. Adler, B. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In Proc. ISOC Network and Distributed System Security Symposium, 2002.

[14] V. Shmatikov. "Probabilistic analysis of an anonymity system", J. Computer Sec., Vol.12, No.3-4, pp. 355–377, 2004

[15] A. Back, U. Moller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems", In Proc. 4th International Workshop on Information Hiding, 2001

[16] A. Serjantov and P. Sewell. "Passive attack analysis for connection-based anonymity systems", In Proc. 8th European Symposium on Research in Computer Security, 2003

[17] P. Samarati, and L. Sweeney, "Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression," Proc. IEEE Symp. Res. in Security and Privacy, 1998

[18] P. Samarati, "Protecting Respondent's Privacy in Microdata Release," IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6, 2001

[19] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems, vol.10, no.5, 2002

[20] M. Gruteser, and D. Grunwald, "Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking", Proc. MobiSys '03, 2003

[21] RFC 3323, J. Peterson, "A Privacy Mechanism for the Session Initiation Protocol", IETF, Nov. 2002

[22] Internet Draft, W. Marshall, et al., "SIP Extensions for Caller Identity and Privacy", IETF, Nov. 2001

[23] M. Castleman "sipanon: A SIP Anonymizer" CS W3998, Columbia University, 2001