Caller Identity Privacy in SIP heterogeneous realms: A practical solution

Giorgos Karopoulos, Georgios Kambourakis, and Stefanos Gritzalis

Info-Sec-Lab Laboratory of Information and Communications Systems Security
University of the Aegean, Samos GR-83200, Greece
{gkar, gkamb, sgritz}@aegean.gr

Abstract

The growing demand for voice services and multimedia delivery over the Internet has raised SIP's popularity making it a subject of extensive research. SIP is an application layer control signaling protocol, whose main purpose is to create, modify and terminate multimedia sessions. Research has shown that SIP has a number of security issues that must be solved in order to increase its trustworthiness and supersede or coexist with PSTN. In this paper our purpose is to address such a weakness, namely the caller identity privacy issue. While some solutions to this problem do exist, we will show that they are inadequate in a number of situations. Furthermore, we will propose a novel scheme for the protection of caller's identity which can also support roaming between different administrative domains. Finally, we provide some performance results, which demonstrate that the proposed solution is efficient even in low-end mobile devices.

1. Introduction

Voice over IP (VoIP) and multimedia services have been offered with lower cost and greater flexibility with the introduction of two signaling protocols: H.323 and Session Initiation Protocol (SIP) [1]. SIP seems to be more popular due to its simplicity and flexibility in comparison to H.323; a fact indicative of this claim is the use of SIP as a signaling protocol in IP Multimedia Subsystem (IMS) of 3G by the 3GPP consortium.

SIP is a text based signaling protocol that uses network elements, the so-called Proxy servers, to route requests from end users, authenticate and authorize requests and provide other services as well. SIP also provides Registrar servers which allow users to upload their current location so that the Proxies can locate them. Other SIP elements include User Agents (UA), which are SIP entities that interact with the user, Redirect servers, which only redirect SIP entities to other entities without processing requests, and Location servers that store users' locations. Typically, Location servers are not SIP entities but they are very important for the operation of the protocol.

The operation of the protocol can better be described with an example; a call between two users. At first, if user O'Brien wants to be reachable, he sends a REGISTER message to the corresponding Registrar. This message contains the user's SIP URI, which is a special type of Uniform Resource Identifier (URI), similar to an e-mail address. When Smith wants to call O'Brien he sends an INVITE message to a Proxy which can be transferred through other Proxies as well in order to reach O'Brien. This message contains both sender and recipient SIP URIs and the current location of Smith so that O'Brien can answer the call. In most cases these messages contain private information about the participating entities and travel through insecure public networks which in turn make privacy preservation a very important and difficult issue. By using such information a malicious party could associate communicating users or maintain user profiles based on their actions.

The difficulty of providing privacy in SIP lies in the fact that the encryption of whole SIP messages is not an option. This holds because specific headers should be readable by intermediate proxies for the proper routing of the messages to their final destination. In this paper we propose a method which protects the privacy of the caller identity without however hiding the IP address of the caller's host or the domain which the caller belongs. We will show that our method preserves the caller's identity privacy even when the messages travel through untrusted Proxies and foreign administrative domains.

The rest of the paper is organized as follows. Section 2 reviews previous work related to private information protection in SIP. In Section 3 we present the problem statement and the solution we propose. In section 4 an analysis of our scheme is provided while in section 5 we present performance results for our scheme compared to standard SIP operation. Section 6 concludes the paper and presents future work.

2. Related work

SIP provides some confidentiality and/or privacy solutions per se; in [1] the discussed mechanisms that are based on cryptography are: S/MIME, SIPS URI and IPsec, while there is also the non-cryptographic solution of "Anonymous" URI. In S/MIME, UAs employ digital certificates for the encryption of SIP headers, bodies or both. While the identities of the participating users can

978-1-4244-2703-1/08/$25.00 ©2008 IEEE

remain secret from others and from intermediate Proxies, these users know each other's identity. SIPS URI is the secure counterpart of a normal SIP URI that dictates the intermediate proxies to use the Transport Layer Security (TLS) protocol in their hop-by-hop communication. According to this solution the privacy of the parties is only protected from other malicious users that eavesdrop SIP messages. Moreover, the utilization of TLS implies that TCP will be used, although Datagram Transport Layer Security (DTLS) [2] is also available, which is nothing more than the TLS counterpart over datagram protocols like UDP. IPsec can also be used for the encryption of messages in a hop-by-hop fashion, especially when the two communicating hops share a common secret. The problem with SIPS URI and IPsec is that since all intermediate Proxies are usually neither trusted nor under the control of a trusted authority, a hop-by-hop encryption cannot be guaranteed in every hop. Finally, when the "Anonymous" URI solution is employed, the caller uses a meaningless URI as an identity and the calling party cannot see the true one. However, this scheme is inadequate if user authentication is required and there is at least one untrusted Proxy in the path between the user and the authenticating Proxy.

The schemes described in [3],[4] are closer related to our proposal than those presented in the previous paragraph and are extensions to the basic SIP protocol. In [3] a new logical role is defined which offers privacy services to end users. According to this scheme, a user sends a normal SIP message (for instance an INVITE) to a Proxy stating that he needs privacy services. The Proxy (or whichever entity offers privacy services) strips the headers that contain private information about the user (like <From> and <Contact>) and replaces them with meaningless values. The entity keeps state information so that it can restore the real values when it receives the relevant response. A quite similar solution is presented in [4]. The user sends a SIP message through a trusted set of Proxies revealing his true identity. When the message is about to leave this trusted domain, the last Proxy withholds the true identity of the user. Similarly to the previous scheme the last Proxy must keep state information in order to route back the responses.

3. Caller Identity Privacy

In this section we will describe our solution for protecting the caller identity in SIP. First we present a general example of SIP's operation and then we propose a scheme that can protect the caller's identity in this environment.

Our aim is to propose a scheme that is practical and can support not only current but future business models as well. In order to make this possible, our scheme is consisted of the following set of requirements:

- The identity of the user should only be known to the user and his Home Proxy. This also means that during the operation of the protocol the identity of the user should only be revealed to his Home Domain and no-one else.
- In order to enable charging the user must be authenticated.
- It should be appropriate for users moving among different administrative domains. This will enable the solution to be used to wireless heterogeneous environments as well.

3.1. Problem statement

We start by presenting a SIP architecture which spans across many different administrative domains. The reason for doing this is to show an as generic as possible architecture and the problems that may arise in such an environment. Our description is so general that applies to either wired or wireless scenarios or a mix of them. We pay special attention on the applicability of our solution to heterogeneous networks which belong to different administrative domains. This is because the next generation of networks will probably be composed of interconnected networks that are not administered by the same provider or by providers that have trust agreements between them. In such an environment where security and/or privacy policies enforcement is not always feasible, measures should be taken so that the user's ID is protected even when it's traveling through untrusted domains.

In Figure 1, O'Brien uses a fixed terminal residing in *miniluv* domain and Smith uses a mobile terminal. Smith's domain is *minitrue* but at the moment he is at a different domain, *minipax*, and wants to contact O'Brien. If Smith's terminal is not aware of its Home Proxy IP address then a possibility is that other Proxies (like Local outbound Proxy) intervene between Smith and *minitrue.org* and between *minitrue.org* and *miniluv.org* as well. Most of the times these Proxies are unknown to Smith and cannot be considered trusted; moreover, Smith has no means to control which Proxies his messages will travel through. This means that if Smith chooses to protect his privacy with TLS, he cannot be aware whether it will be used in all hops and so his identity hiding is not always assured. What is needed in this case is a solution that is not based on TLS (or other hop-by-hop encryption method) and selectively makes Smith's identity known only to trusted entities, while hiding it from untrusted ones.

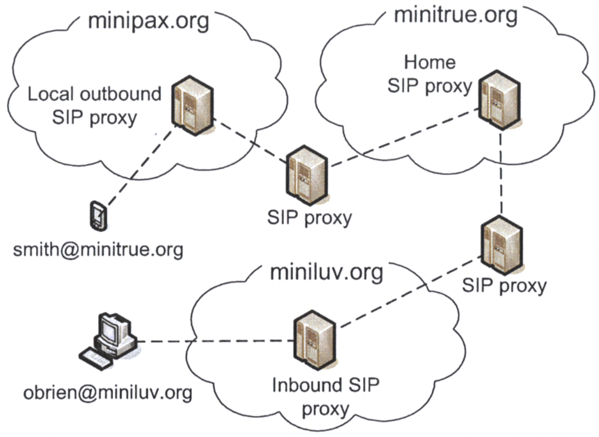

Figure 1. Multidomain SIP architecture

3.2. Our solution

According to the proposed scheme, caller identity hiding is supported even when untrusted Proxies reside between trusted parties. In order to fulfill this requirement we use asymmetric cryptography and encrypt the caller's identity with the Proxy's public key so that only this trusted entity can recover it. At the same time, everybody else (including other users and Proxies) has access only to the encrypted form of the identity.

Without loss of generality, we employ an example where someone wants to place a call to another user. We start by examining the headers of the respective SIP message, e.g. an INVITE sent from Smith to O'Brien (other SIP messages have similar headers):

```
INVITE sip:obrien@miniluv.org SIP/2.0
Via: SIP/2.0/UDP 195.251.161.144:5060;
branch=z9hG4bK74b43
Max-Forwards: 70
From: Smith <sip:smith@minitrue.org>;
tag=9fxced76sl
To: O'Brien <sip:obrien@miniluv.org>
Call-ID: 3848276298220188511@minitrue.org
CSeq: 1 INVITE
Contact: <sip:smith@minitrue.org>
Content-Type: application/sdp
Content-Length: 151
```

As it can easily be seen in the above message, particular headers reveal private information about the two communicating parties. In this paper our concern is the protection of the caller identity so we will not consider the called party's privacy. The headers that reveal information about the caller, i.e. Smith, are: <Via> header reveals the caller's host IP address, <From> and <Contact> reveal the SIP URI (which is composed from the user's identity followed by his home domain) and <Call-ID> reveals the domain where the caller belongs (in this case *minitrue.org*).

We must stress out here that our purpose is to protect only the identity of the user and not all the information about him like his host's IP and the domain he belongs. To do so, we strip whichever information is not necessary and use encryption for the rest. More specifically:

- we leave <Via> field's value as is, because it only reveals the IP address of the host
- <Contact> field's value is replaced with the IP address of the caller's host. End users' IP addresses usually are not static so eavesdroppers cannot easily relate it with the permanent ID of the user
- the display name in <From> field ("Smith" in our example) is stripped or replaced by the string "Anonymous", and
- the user ID part of <From> field (i.e. "smith" in "smith@minitrue.org") is encrypted using asymmetric cryptography with the public key of the Home Domain's Proxy. As it is obvious we propose a scheme that rather relies on pseudonymity than anonymity [5]. If the same pseudonym is always used then the user can be "profiled" and his movement (in case of a mobile user) can be easily tracked. For this reason a padding scheme (like the Optimal Asymmetric Encryption Padding – OAEP one [6] for RSA) should be used so that the resulting pseudonym is different every time.

The resulting message is shown below; in this example the hexadecimal representation is used for the encrypted part of the URI.

```
INVITE sip:obrien@miniluv.org SIP/2.0
Via: SIP/2.0/UDP 195.251.161.144:5060;
branch=z9hG4bK74b43
Max-Forwards: 70
From: <sip:0AEE5F83...129F32@minitrue.org>;
tag=9fxced76sl
To: O'Brien <sip:obrien@miniluv.org>
Call-ID: 3848276298220188511@minitrue.org
CSeq: 1 INVITE
Contact: 195.251.161.144
Content-Type: application/sdp
Content-Length: 151
```

If authentication is not required then the most practical and effective solution would be the employment of "Anonymous" URI in <From> header. However, in a real world environment the most probable case is that the user must be authenticated in order to be charged for the services he receives. If caller identity privacy is also a requirement then the existing schemes are not adequate as we have already showed. In this paper we only consider Digest authentication [7]. In the following we will present an example where both the Local outbound Proxy and Home Proxy require Smith to authenticate in order to receive their services. We assume that Smith has a different set of credentials for each of the two domains and he is willing to present each of the two identities he possesses only to the corresponding domain.

Caller identity privacy during the authentication process can be assured in a similar way as in the previous example. When the INVITE message is received, the Local outbound Proxy responds with a 407 Proxy Authentication Required message. Smith sends back a new INVITE where he encrypts the username used in <Proxy-Authorization> field with the public key of the Local outbound Proxy as shown below, while the user ID part of <From> field again is encrypted with the public key of Home Proxy. The different user IDs used here are in accordance with [1] and reveal each ID only to the intended Proxy.

```
INVITE sip:obrien@miniluv.org SIP/2.0
Via: SIP/2.0/UDP 195.251.161.144:5060;
branch=z9hG4bK74b43
Max-Forwards: 70
From: <sip:0AEE5F83...129F32@minitrue.org>;
tag=9fxced76sl
To: O'Brien <sip:obrien@miniluv.org>
Call-ID: 3848276298220188511@minitrue.org
CSeq: 1 INVITE
Proxy-Authorization: Digest
username="38A8F347...0EA19A98", algorithm=MD5,
realm="minitrue.org", nonce="1dea4387...00f4e5da",
qop="auth", opaque="5e7734afdb981200",
response="ffa1e3...8756ee", nc=00000001,
cnonce="abcdefghi"
Contact: 195.251.161.144
Content-Type: application/sdp
Content-Length: 151
```

The Local outbound Proxy decrypts Smith's username and completes the authentication process and, if it is successful, it forwards the INVITE to Smith's Home Proxy. The Home Proxy also completes authentication in the same manner. After that, the initial INVITE message is forwarded to the Inbound Proxy which sends it to O'Brien. As we can see no untrusted entities involved in the protocol (including O'Brien) are aware of Smith's identity. When O'Brien answers the call, his response travels all the way back to *minitrue.org* where the Proxy deciphers <From> header to discover the recipient of the message.

While the usefulness of our scheme is proven through examples, this does not limit its generality. The same procedure would be followed if, for instance, there were Registrars instead of Proxies and REGISTER messages instead of INVITEs.

4. Analysis

As it has already been discussed, our scheme protects only the user ID part of the SIP URI and leaves the host's IP address and user's home domain name unprotected. In fact there are solutions for the protection of this information like those reviewed in [8]; however, this is out of the scope of this paper. Another reason for not choosing to adopt such techniques is because the solution we provide makes a trade off between security and practicality. Our scheme preserves the secrecy of the caller's identity while at the same time remains simple, scalable and easy to deploy.

Our mechanism has a number of advantages over the existing solutions discussed previously in Section 2:

- It can be used in a SIP network consisting of any kind of Proxies, either stateless or stateful. In other solutions the Proxy server replaces the identity of the user with some other string and needs to be stateful in order to store the correspondence between those two values. In our scheme the SIP Proxy only needs to decipher the value in <From> field. It is worth noting that stateful proxies are often the victims of Denial of Service (DoS) attacks.
- The changes needed in the existing infrastructure are reasonable. There is no need for new (hardware or software) entities; however, every Proxy needs to have a digital public key certificate and be able to decipher the encrypted values. So, some sort of rudimentary Public Key Infrastructure (PKI) is needed, although it must be noted that digital certificates will not be issued for end users.
- It can be used in cases where the exchange of messages among different administrative domains is necessary. It only relies on the trust between the user and his Home Domain's Proxy and because the identity information is encrypted it can be transmitted through other (possibly unknown and untrusted) Proxies.
- It is applicable in wireless scenarios and especially during the handover procedure when SIP is used for mobility support [9]. Moreover, as we will show, the delay of creating an INVITE message (or re-INVITE in the case of a handover) is minimal and does not require time consuming procedures like a TLS handshake does.

One possible disadvantage of our scheme stems from the fact that it is not based on anonymity but on pseudonymity. This means that the called party can always return the call by using the caller's pseudonym, something that is avoided in anonymous schemes.

From the above analysis of our scheme and comparing it to other similar solutions we can come to a conclusion as when the proposed scheme is more suitable to use. Our mechanism seems to be the only solution when multiple different administrative domains are involved within the route of the messages. It also seems to be more suitable to situations where the caller is not sure which or how many Proxies are involved in the transmission of SIP messages to their final destination. A closer look at the details of our proposal shows that user and/or terminal mobility can further improve user's privacy. This is because in the first case, i.e. user mobility, the user makes calls from different terminals (for instance from home, a corporate network, a mobile network and so on) which have different IP addresses and this makes it more difficult to infer the user's identity from an IP address. The same applies in the second case when a roaming user operates a single terminal and roams between different wireless networks; during this movement his IP address keeps changing with the choice being made by the visited realm. Even when the user uses a steady host in most cases its IP address is dynamically assigned (although in practice it may not change frequently). In any case our aim is not to provide a complete privacy solution for SIP but rather an adequate and lightweight one which could scale well in demanding situations like handovers between heterogeneous networks and/or networks that belong to different administrative domains.

5. Performance

The performance of the proposed scheme was evaluated in a testbed and the results are depicted in this section. Our purpose here is not to evaluate SIP's performance in general but to show the performance penalty imposed by our method compared to standard SIP transactions. In order to conduct our experiments we constructed an experimental network architecture which comprises from the following elements:

- one SIP proxy server with an Intel Pentium 4 Hyper-Threading CPU at 3.2 GHz and 1 GB of RAM. Our server connects to the network through a Broadcom NetXtreme Gigabit Ethernet card. The SIP proxy software is based on SIP Express Router (SER) [10] version 0.9.6.
- one SIP UA (high-end UA) on a desktop PC with an Intel Pentium 4 Hyper-Threading CPU at 2.6 GHz and 512 MB of RAM, which also connects to the network through a Broadcom NetXtreme Gigabit Ethernet card. The software used for measuring SIP server's delay is a modified version of SIPp 3.0 [11]. The software used for measuring client's request preparation delay is based on Twinkle SIP softphone version 1.1 [12].
- one SIP UA (low-end UA) on a laptop machine which incorporates an AMD Mobile Athlon 4 CPU at 1.2 GHz and 256 MB of RAM. For the purposes of our experiments, the laptop's CPU was downgraded from 1.2 GHz to 500 MHz with the use of Powersave daemon version 0.10.15 in order to have similar capabilities as today's handheld and mobile devices. Its network interface is an Intel PRO/100 Series Mini PCI NIC Ethernet card supporting speeds up to 10 Mbps. The software used for measuring client's delay is also based on Twinkle SIP softphone version 1.1.

All the employed machines are based on the same OS which is SuSE Linux 10.0 with kernel version 2.6.13-15-smp, and gcc version 4.0.2. SER was supported by MySQL version 5.0.45-community during the authentication procedure. A single 1024 bit RSA digital certificate has been issued for the Proxy server and the public key has been transferred to the UAs. The measurements where conducted in a 100 Mbps non congested LAN.

We have made the following modifications to the initial versions of the software used:

Twinkle: Our modified Twinkle first reads Proxy's public key from a local certificate file (.pem) and then encrypts the user ID using RSA with OAEP encoding.

SER: Our modified SER uses its private key to decrypt the user ID, processes the request and forwards the message with the original encrypted user ID. When Digest authentication is used it also decrypts the username of the UA.

SIPp: SIPp creates SIP messages based on an XML file that describes a scenario. While encrypted SIP URIs are parsed correctly, we had to modify SIPp in order to parse long usernames (in our case 256 characters). When a 407 Proxy-Authorization request is received, SIPp's response includes the encrypted forms of the user ID and the username used for authentication.

We have tracked and logged results based on two distinct scenarios:

Client delay: We measured the time required for a UA to construct an INVITE request. Moreover, for comparison purposes, we recorded measurements when our scheme is used and when it is not, both on high-end and low-end UAs. The measured request creation phase constitutes from the preparation of all SIP headers including the encryption of user ID when our scheme is utilized.

Server delay: We measured the time required for a SIP Proxy server with different queue sizes to serve a request. The scenario was executed two times, one using standard SIP and one using our proposal; in both cases SER operates in stateful mode. For each queue size the call rate is automatically adjusted by SIPp. The measured time starts when an INVITE is send and ends when a "100 Trying" is received by SIPp; this means that the user has been authenticated and his call is been forwarded. The delays included are: the parsing of the unauthenticated INVITE by SER (for our proposal SER decrypts UA's URI), the digest response preparation time by the UA (no encryption takes place here; the encrypted values used are hardcoded in SIPp's scenario file), the parsing of UA's response (for our proposal involves the decryptions of UA's URI and username) and finally the corresponding roundtrip times.

For the first scenario we have taken measurements with four different configurations. For each configuration we have measured the delay of the preparation of a single INVITE message 1,000 times. These configurations are:

1. High-end UA with no caller identity privacy
2. High-end UA with caller identity privacy
3. Low-end UA with no caller identity privacy
4. Low-end UA with caller identity privacy

Table 1 shows the results for each of the 4 different configurations. The observation of the table reveals that when our scheme is in use the INVITE preparation delay is almost 4 times higher compared to standard SIP and this is obviously due to cryptographic operations involved. However, all delays measured are in msecs with a maximum of 8.14 msecs, meaning that there is no perceived delay by the end user. Also standard deviation of all values remains low, showing that their majority is spread near the mean delay.

Table 1. Request preparation delay

Configuration	Delay (msec)			Standard
	Mean	Max	Min	dev at n
1	0.16	1.34	0.14	0.07
2	0.61	3.01	0.55	0.13
3	0.38	6.11	0.31	0.2
4	1.6	8.14	1.36	0.26

The distribution of measured delays is presented for each configuration in Figures 2, 3, 4 and 5 correspondingly. The X axis represents the INVITE preparation delay in msecs, while Y axis shows the number of occurrences of each delay. Note that in each diagram some of the maximum values were not included in order to make these diagrams more readable.

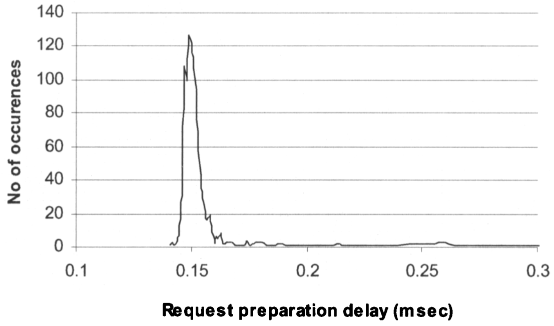

Figure 2. High end UA with no caller identity privacy

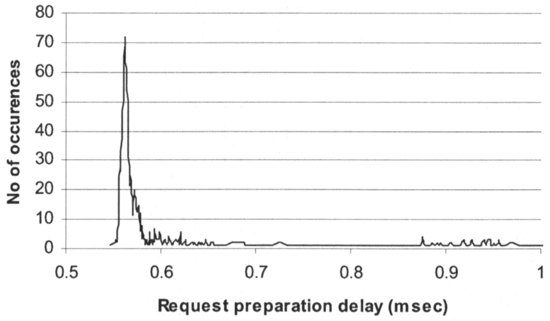

Figure 3. High end UA with caller identity privacy

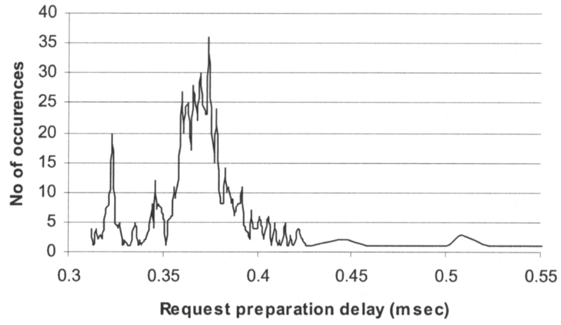

Figure 4. Low end UA with no caller identity privacy

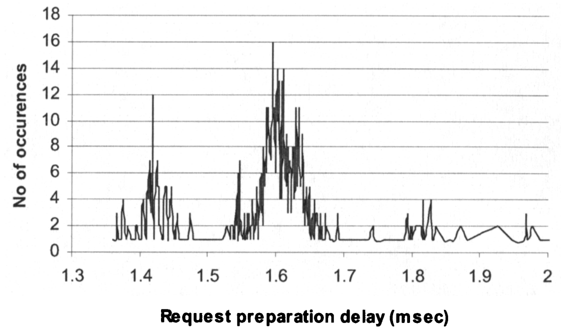

Figure 5. Low end UA with caller identity privacy

During the execution of the second scenario we measured the mean server response times for different queue sizes. For each queue size we computed the mean response time of 1,000 authentication handshakes. For each different scheme, server's queue is populated with similar requests, e.g. standard SIP messages for measuring standard SIP's response delays, and privacy enhanced messages for measuring our scheme. All measures were taken after leaving a 20 sec warm-up period for SER.

Table 2 shows the results for the second scenario. From these results we find that there is a significant difference in response delays between standard SIP and our proposal. However, these results are based on the assumption that in the first case we only have standard SIP requests while in the second case only our modified requests. In a more realistic scenario (where probably privacy will be preserved with some additional cost) the requests will be mixed and the performance penalty will be decreased.

Table 2. Server response delay

Server queue size (calls)	Standard SIP (msec)	Proposed scheme (msec)	Difference (msec)
250	89.18	606.94	517.76
500	91.23	658.1	566.87

Server queue size (calls)	Standard SIP (msec)	Proposed scheme (msec)	Difference (msec)
750	93.6	740.81	647.21
1000	103.78	699.44	595.66
1250	103.18	773.56	670.38
1500	110.19	842.89	732.7
1750	113.19	960.92	847.73
2000	114.52	894.73	780.21
2250	116.75	1035.18	918.43
2500	122.08	1108.44	986.36

Figure 6 depicts the mean server response delays for different server queue sizes. The X axis represents the size of the queue, while Y axis shows the mean response delay computed for each size in msecs.

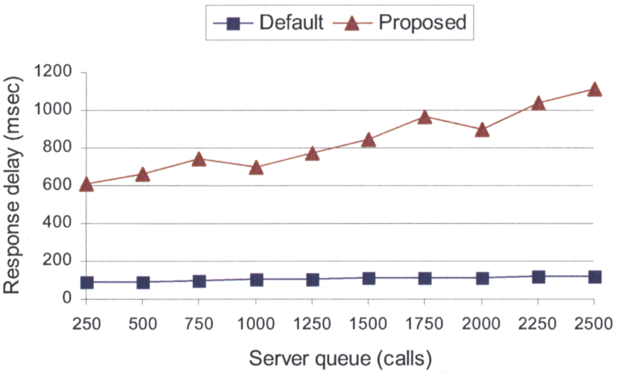

Figure 6. SIP and caller identity privacy enhanced SIP server response delays

6. Conclusions

We have presented a novel scheme for caller identity protection in SIP. Our proposal proved to have certain advantages over similar solutions in scenarios where multiple administrative domains and untrusted SIP proxies are involved. We evaluated our mechanism by conducting some performance tests which reveal that even if the request preparation delay is almost 4 times higher, it is insignificant in terms of service time. Our tests also showed a performance decrease for SIP Proxies; however, the user perceived delay increases significantly only when the Proxy is heavily congested.

Since our proposal is not a total privacy solution for SIP, our future work includes further research for providing more options to end-users to protect their privacy when using SIP. Our aim is to propose practical solutions that protect private information from exposure, while at the same time come with little additional cost in terms of time delay and minor modifications to underlying infrastructure.

7. Acknowledgement

This paper is part of the 03ED375 research project, implemented within the framework of the “Reinforcement Programme of Human Research Manpower” (PENED) and co-financed by National and Community Funds (25% from the Greek Ministry of Development-General Secretariat of Research and Technology and 75% from E.U.-European Social Fund).

8. References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol, RFC 3261, June 2002.

[2] E. Rescorla and N. Modadugu. Datagram Transport Layer Security, RFC 4347, April 2006.

[3] J. Peterson. A Privacy Mechanism for the Session Initiation Protocol (SIP), RFC 3323, November 2002.

[4] C. Jennings, J. Peterson, and M. Watson. Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, RFC 3325, November 2002.

[5] A. Pfitzmann and M. Köhntopp, “Anonymity, Unobservability and Pseudonymity - A Proposal for Terminology,” Designing Privacy Enhancing Technologies: Proc. International Workshop Design Issues in Anonymity and Observability, LNCS, vol. 2009, pp. 1-9, Springer-Verlag, Berlin, 2000.

[6] M. Bellare, P. Rogaway. “Optimal Asymmetric Encryption - How to encrypt with RSA.” Extended abstract in Advances in Cryptology - Eurocrypt ‘94 Proceedings, LNCS, Vol. 950, A. De Santis (ed.), Springer-Verlag, 1995.

[7] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication, RFC 2617, June 1999.

[8] S. Gritzalis, “Enhancing Web Privacy and Anonymity in the Digital Era”, Information Management and Computer Security, vol. 12, No. 3, pp. 255-288, 2004, Emerald.

[9] H. Schulzrinne, and E. Wedlund, “Application-layer mobility using SIP,” SIGMOBILE Mobile Computing and Communications Review, vol. 4, No 3, pp. 47-57, July 2000.

[10] SIP Express Router (SER), free, open source SIP server, available at http://www.iptel.org/ser

[11] SIPp, performance testing tool for SIP, available at http://sipp.sourceforge.net

[12] Twinkle SIP softphone, available at http://www.twinklephone.com