Computer Standards & Interfaces 33 (2011) 301–314

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

PrivaSIP: Ad-hoc identity privacy in SIP

Giorgos Karopoulos *, Georgios Kambourakis, Stefanos Gritzalis

Info-Sec-Lab: Laboratory of Information and Communication Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Samos GR-83200, Greece

ARTICLE INFO

Article history:
Received 12 November 2008
Received in revised form 24 June 2010
Accepted 6 July 2010
Available online 17 July 2010

Keywords:
Privacy
SIP
Security
Heterogeneous networks
Performance evaluation
Multimedia

ABSTRACT

In modern and future networks that belong to different providers, multimedia protocols will have to operate through multiple domains. In such an environment security is considered a crucial parameter; this is true especially for privacy since not all domains can be considered trusted beforehand in terms of personal data protection. Probably the most promising protocol for multimedia session management is SIP. While SIP is popular and a lot of research has been conducted, it still has some security issues, one of which is related to privacy and more particularly the protection of user identities (IDs). In the general case everybody can reveal the communicating parties IDs by simply eavesdropping on the exchanged SIP messages. In this paper we analyze the lack of user ID protection in SIP and propose two solutions; in the first the ID of the caller is protected while in the second both IDs of the caller and the callee are protected. Our work also includes performance results and extensive comparison with similar methods. The most significant advantage of our method is that it can assure user ID protection even when SIP messages are transmitted through untrusted SIP domains before reaching the Home Domain of the user or another trusted domain. Moreover, it does not require from the SIP Proxy server to maintain state information for exchanged SIP requests and respective responses.

1. Introduction

Multimedia is an application class with great importance in today's networks, no matter whether these are wired or wireless. In fact, it is important that multimedia delivery is based on interoperable protocols so that converged (and possibly heterogeneous) networks can offer uninterrupted services. It is expected that the next generation of wireless networks, namely 4G, will be based on IP, realizing an all-IP architecture. It is obvious at this point that such IP based networks will be fully compliant with wired networks and the Internet with no need for gateways or other translation means. In such an environment the multimedia deliverance will be possible even when users move or change between networks with different access layer technologies. This type of roaming can be realized with schemes like those proposed in [21].

One of the most important protocols supporting multimedia services is Session Initiation Protocol (SIP) [1]. SIP is an application layer control signaling protocol responsible for the creation, modification and termination of multimedia sessions. One of the facts that show the significance of SIP is that 3GPP consortium [2] chose it to be the multimedia management protocol of 3G networks multimedia subsystem (IP Multimedia Subsystem—IMS). Since SIP is an application layer protocol, it can transparently operate over any type of network; furthermore, it also has the ability to support application layer handovers when a lower layer handover occurs [3].

SIP has been a protocol which has received extensive attention and part of the research has shown that it suffers from security issues [4] some of which have already been solved [4,5,22]. In this paper we focus on privacy and more specifically on the protection of user IDs that normally are publicly available to anyone who eavesdrops on the underlying network. While there are some solutions for protecting the privacy of end users, these are not adequate in certain environments compared to the proposed schemes.

The existence of several overlapping networks in 4G will lead to a plethora of choices between different network providers for the user. Taking into account that multimedia content providers could be other than the network providers it is obvious that each user has to communicate with different administrative domains. These domains will not always be known or trusted beforehand so the users must be very careful when revealing their IDs to such foreign domains. The only viable assumption that can be made in such environments is that only the Home Domain of the user can be considered trusted.

In this paper we present two protocols that protect the IDs of communicating users regardless of the number or the level of trust of domains that reside between them. Moreover, our protocols operate in an ad-hoc manner, requiring no prior trust agreements between the user and his Home Domain other than the possession of the digital certificate of the respective SIP Proxy server. We also provide performance analysis of our methods through an appropriate testbed and compare our results with standard SIP that provides no ID privacy.

* Corresponding author. Tel.: +30 22730 82246; fax: +30 22730 82009.
E-mail addresses: gkar@aegean.gr (G. Karopoulos), gkamb@aegean.gr (G. Kambourakis), sgritz@aegean.gr (S. Gritzalis).

0920-5489/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2010.07.002

Furthermore, we review existing solutions in SIP privacy and compare them with our own proposals.

Next section starts by presenting the ID privacy issues of SIP in more detail. In this section the problem statement is given and two solutions are proposed, namely PrivaSIP-1 and PrivaSIP-2. In Section 3 we provide time delay measurements of our schemes in comparison to standard SIP. Section 4 defines different privacy levels for SIP IDs while Section 5 analyzes the existing solutions to SIP privacy issues. In Section 6 we theoretically compare our schemes with existing solutions based on several defined criteria. In Section 7 the outcome of the above comparison is discussed, outlining the most significant points observed. Section 8 summarizes the contribution of this paper compared to previous work, while Section 9 concludes the paper and gives some directions for further research.

2. SIP identity privacy

In this section we will describe the ID privacy issue and our solutions for protecting user IDs in SIP. The first scheme, which was previously presented by the authors in [6,20], offers caller's ID privacy while our second scheme protects both caller's and callee's IDs [20].

2.1. Problem statement

We start by presenting a SIP architecture which spans across many different administrative domains. The reason for doing this is to demonstrate an as generic as possible architecture and describe more clearly the problems that may arise in such an environment. Our analysis 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, in terms of access technology, networks which belong to different administrative domains. This is because the next generation of networks, also known with the term 4G, will probably be composed of interconnected networks that may not be administered by the same provider or by providers that have trust agreements between them. In such a many-to-many fashioned environment where security and/or privacy policies enforcement is not always feasible, measures should be taken so that users IDs are protected even when they are traveling through untrusted domains. Without loss of generality, for the remainder of the paper we employ an example of a Voice-over-IP (VoIP) call between two users. However, the proposed solutions apply as is to other types of multimedia sessions as well.

In Fig. 1, O'Brien uses a fixed terminal residing in miniluv domain and Smith uses a mobile terminal. Smith's Home Domain is minitrue but at the moment he roams to a different domain, namely minipax, and wants to contact O'Brien. If Smith's terminal is not aware of its Home SIP Proxy's IP address then a possibility is that other Proxies (like Local outbound Proxy) intervene between Smith and minitrue.org as well as between minitrue.org and miniluv.org. Most of the times these SIP Proxies are unknown to Smith and cannot be considered trusted; moreover, Smith has no means to control which Proxies his messages will travel through. Considering ID privacy, if for example Smith chooses to protect his privacy with Transport Layer Security (TLS), he cannot be aware whether it will be used in all hops and therefore his ID 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 ID known only to trusted entities, while hiding it from untrusted ones. The answer to this problem is given by our first scheme, namely PrivaSIP-1.

Considering the previous example the information that is revealed to third parties is that a user from minitrue.org domain has a conversation with O'Brien from miniluv.org. A more effective, in terms of privacy, scheme could also protect O'Brien's ID so that the only information available to others would be that a user from minitrue.org has some sort of communication with a user from miniluv.org. The way to accomplish this is described in PrivaSIP-2, our second scheme.

There are a number of malicious acts associated with the lack of user ID privacy. The first and more obvious one is that everybody can have access to information regarding who is communicating with whom. If this information is systematically gathered then a certain user can be profiled, based on VoIP calls and other multimedia usage. When SIP URIs are made available then a possible attack is also Spam over IP Telephony or SPIT [7] which is similar to e-mail spam. Another consideration is that the movement of a specific user can be tracked by observing the transmitted IDs. This can happen when a mobile user handovers between different networks and transmits his ID in order to transfer the existing session to the new network. This can also be the case when session mobility is supported and a certain user continues using a session but changes between different devices, either mobile or not.

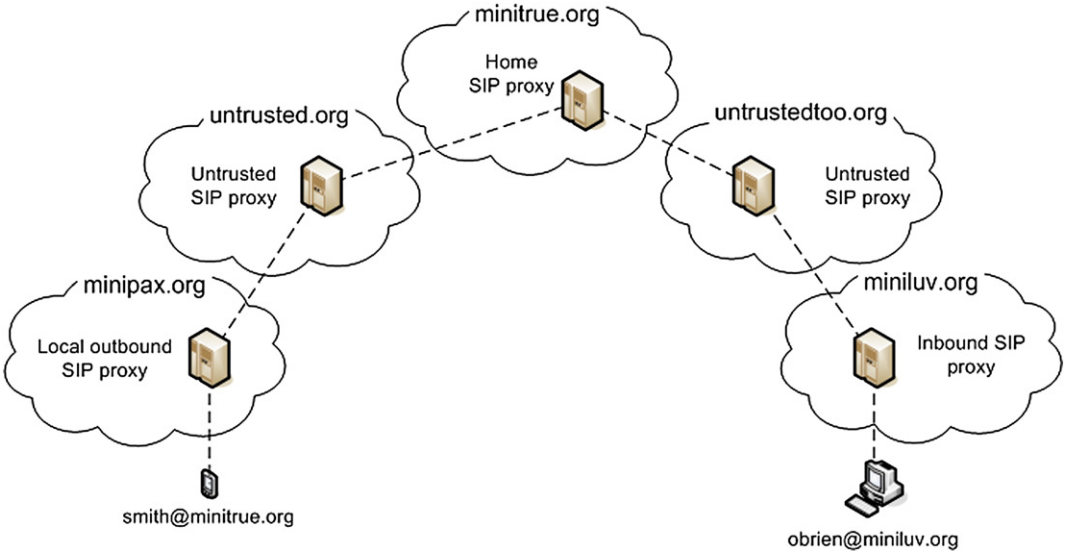

Fig. 1. Multidomain SIP architecture.

2.2. PrivaSIP-1: caller identity privacy

According to the scheme we have proposed in [6,20], caller ID hiding can be supported even when untrusted SIP Proxies reside between trusted parties, like in the example shown in Fig. 1. In order to fulfill this requirement we use asymmetric cryptography and encrypt the caller's ID with the Home Proxy's public key so that only this trusted entity can recover it. At the same time, everybody else (including the callee, other users and Proxies) has access only to the encrypted form of the ID.

We start by examining the headers of a SIP message used for placing a call, 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 scheme our concern is the protection of the caller ID, while in the one analyzed in the next section we will show how we can protect called party's privacy as well. 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 ID followed by his Home Domain name) 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 ID of the user and not all the information about him like his host's IP and the domain he belongs. In fact, such information is necessary since the callee must know the caller's IP in order to eventually establish a peer-to-peer media session. Moreover, third party Proxies must be aware of caller's Home Domain in order to forward the message to the right SIP Proxy. Also, our mechanism does not aim at protecting the confidentiality of whole messages or provide message integrity; such requirements should be met by utilizing other mechanisms. The solution we propose is to 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 SIP Proxy. As it is obvious we propose a scheme that rather relies on pseudonymity than anonymity [8]. 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 [9] 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 (see Section 5.4). 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 ID privacy is also a requirement then the existing schemes are not adequate as we will show in subsequent sections. In this paper we only consider Digest authentication [10] which is the standard way of authenticating users in SIP environments.

In the following we will present an example where both the Local outbound SIP 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 IDs he possesses only to the corresponding domain. Since Smith has credentials from both domains it means that he has some kind of agreement with each one of them, so he is aware of what kind of private information he presents to each domain. The key point here is that the caller has the choice to present private information only to selected domains minimizing the number of entities that posses this information. Caller ID 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 is encrypted with the public key of Home Proxy. What we must note here is that this asymmetric encryption process does not imply in any way that it supports user authentication. This task is conducted with the utilization of Digest authentication. 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="38A8-
F347...0EA19A98", algorithm=MD5, realm="minitrue.
org", nonce="1dea4387...00f4e5da", qop="auth", opa-
que="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 ID. When O'Brien answers the call he uses the same encrypted headers and 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 SIP Registrars instead of Proxies and REGISTER messages instead of INVITEs.

2.3. PrivaSIP-2: total identity privacy

In this section we further improve the aforementioned scheme so that it also preserves called party's ID as well, as proposed in [20]. Our purpose is to have two alternative schemes each one used in different situations depending on imposed privacy requirements. In Section 4 we will make a short list of privacy requirements with regard to user's ID and discuss which of these requirements are met by different privacy preserving solutions in Section 5.

In order to present the inner workings of our second scheme we will use the same example as in the previous section. The protection of callee's ID is achieved by a similar mechanism as the caller's ID with the use of asymmetric cryptography; more specifically we encrypt the ID with the public key of callee's Home Domain SIP Proxy. It must be noted here that we also protect caller's ID as shown in the previous section.

As we already presented in the previous section some SIP headers of an INVITE message must be scrambled to protect the ID of the caller. Apart from these headers, in this scheme we also protect <To> field which exposes callee's ID. The resulting message is shown below:

```
INVITE sip: 73D8A9F7...BC09E1A1@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: <sip:73D8A9F7...BC09E1A1@miniluv.org>
Call-ID: 3848276298220188511@minitrue.org
CSeq: 1 INVITE
Contact: 195.251.161.144
Content-Type: application/sdp
Content-Length: 151
```

What applies for user authentication in our first scheme also applies here. The caller can hide both his ID and Digest password, while also the callee's ID is protected from third parties. The procedure that is followed is the same as presented in the previous section except that when an INVITE is forwarded to the Inbound SIP Proxy, the <To> field is decrypted and subsequently send to O'Brien. When O'Brien responds back he uses the same encrypted headers so that the privacy enhanced SIP message is routed appropriately.

3. PrivaSIP service time measurements

The performance of the proposed schemes for both the client and the server was evaluated in a properly designed testbed and the results are depicted in this section. It is well known that security or privacy mechanisms come always at a cost. However, apart from the effectiveness and robustness of the proposed mechanism, the key question in every case is if that cost is affordable. So, our intension here is not to evaluate SIP's performance in general but to determine the performance penalty imposed by our methods compared to standard SIP transactions. In the related work section further down we extensively discuss all known schemes that could be used for providing some sort of privacy in SIP. However, we do not compare the performance of these methods with that of PrivaSIP. The chief reason for not doing so is that each scheme presents different qualities, and each of them is useful under a specific context, irrespective of the performance penalty one might impose. For example, when the user ID must be protected and authentication is also a requirement, then PrivaSIP is the only viable solution; when authentication is not a requirement, then Anonymous URI is the right choice. Also, other solutions either do not provide enough or assured privacy (IPsec, SIPS URI/TLS) or do not protect privacy during authentication (Privacy mechanism) or do not support authentication at all (S/MIME, Anonymous URI).

We have already presented some initial results for our first method in [6]. However, here and in [20] we present several new results for our first method's server delay. This is essential since in [6] we measured only a part of a SIP call while here we measure its overall time, i.e., from its initiation until the ringing phase. Another reason is that part of the testbed used in [6] has been substantially changed in the current work to be more realistic. The difference between the two SIP call flows (roundtrips) is depicted in Fig. 2. More specifically, the delays we presented in our previous work measure the time from the

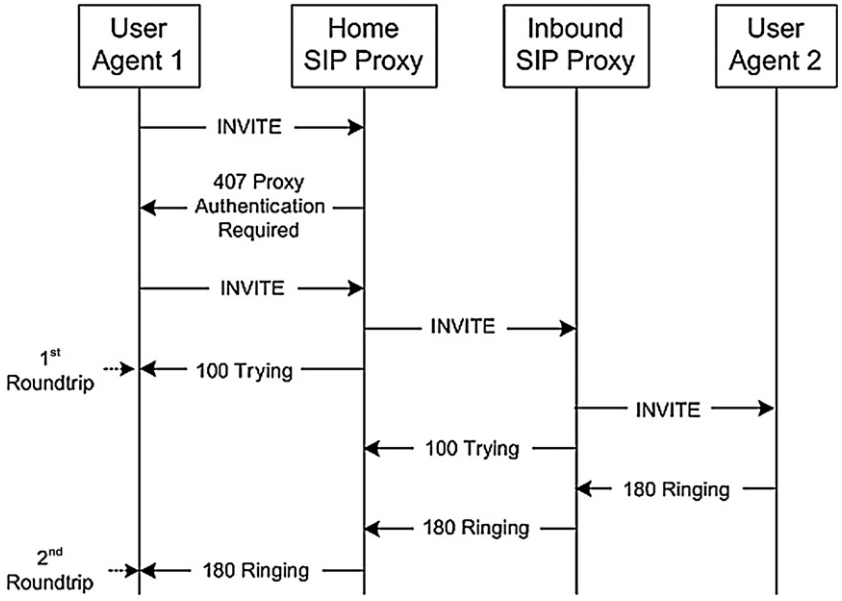

Fig. 2. Standard SIP call flow.

initiation of the call until the end of the first roundtrip, while here we measure the delay until the second roundtrip. The reason we do this is that in our previous work we only modified our Home SIP Proxy in order to cope with our cryptographically protected messages, while all other network elements are based on standard SIP. So there was no need to take the whole call until the second roundtrip into consideration. Here, on the other hand, both SIP Proxies were modified and we were forced to rerun our experiments in order to be able to compare our schemes to each other.

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

1. 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. The measured request creation phase constitutes from the preparation of all SIP headers including the encryption of user ID when our scheme is utilized. We have measured delays using a "low-end client" as well as a "high-end client" so that we could investigate what is the impact of our method on different hardware configurations. This scenario runs only on clients and does not involve any network interaction since we only measure the INVITE preparation delay.
2. Server delay. We measured the time required for a SIP Proxy server with different queue sizes to serve a request. The scenario was executed three times, one using standard SIP, one using our first scheme (PrivaSIP-1) and one using our second scheme (PrivaSIP-2). For each queue size the call rate is automatically adjusted by SIPp [13]. The measured time starts when an INVITE is send and ends when a "180 Ringing" is received by SIPp; this means that the user has been authenticated and his call has reached the intended recipient. It must be noted here that we take the worst case scenarios; all SIP URIs and digest usernames are computed each time they are needed, and no party stores call state information. The delays included are:

   - the parsing of the unauthenticated INVITE by Home Proxy (for our schemes SER [11] decrypts caller's URI),
   - the digest response preparation time by the caller's 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 schemes this involves the decryptions of UA's URI and username),
   - the parsing of INVITE by Inbound Proxy (for PrivaSIP-2 only, this involves the decryptions of callee's URI) and finally
   - the respective network delays.

In order to conduct our experiments we constructed an experimental network architecture which comprises from the following elements (also summarized in Table 1):

- one low-end 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, which is part of the machine's Operating System (OS). This enabled us to have similar capabilities as today's handheld and mobile devices. The laptop's network interface was not used since it ran only the client scenario as a "low-end UA". The OS of this machine is SuSE Linux 10.0, kernel version 2.6.13-15-smp, with gcc version 4.0.2, and the software used for measuring client's delay is based on Twinkle SIP softphone version 1.1 [12].
- one desktop PC with an Intel Pentium 4 Hyper-Threading CPU at 2.6 GHz and 512 MB of RAM, which also does not utilizes its network card since it is the "high-end User Agent (UA)" for measuring client delay. The OS of this machine is SuSE Linux 10.0, kernel version 2.6.13-15-smp, with gcc version 4.0.2 and the software used for measuring client's delay is based on Twinkle SIP softphone version 1.1.
- one desktop with a dual-core Intel Pentium 4 CPU at 3 GHz and 1 GB of RAM which plays the role of "User Agent 1" in Fig. 2. This machine connects to the network through a Broadcom NetXtreme Gigabit Ethernet card. Its purpose is to make multiple calls to User Agent 2 through the two Proxies so that we can measure the delay of each request when the Proxies have queue sizes of certain length. This is realized with the use of SIPp 3.0 in client mode which automatically adjusts the call rate so that a stable queue size is maintained. This machine's OS is openSuSE Linux 10.3, kernel version 2.6.22.18-0.2, with gcc version 4.2.1.
- one PC with a dual-core AMD Athlon X2 64 CPU at 1.9 GHz and 2 GB of RAM which plays the role of "Home SIP Proxy" in Fig. 2. This machine connects to the network through a Realtek RTL8102E Fast Ethernet 100 Mbps network card. The SIP proxy software is based on SIP Express Router (SER) version 0.9.6 supported by MySQL version 5.0.45-community during the authentication procedure. This machine's OS is openSuSE Linux 11 (32-bit version), kernel version 2.6.25.16-0.1 with gcc version 4.3.
- one desktop PC with a dual-core Intel Pentium 4 CPU at 2.8 GHz and 1 GB of RAM, which connects to the network through a Broadcom NetXtreme Gigabit Ethernet card and is used as the "Inbound SIP Proxy" in Fig. 2. The OS of this PC is openSuSE Linux 11, kernel version 2.6.25.16-0.1 with gcc version 4.3. The SIP proxy software is based on SER version 0.9.6.
- one desktop with a dual-core Intel Pentium 4 CPU at 2.6 GHz and 512 MB of RAM which plays the role of "User Agent 2" in Fig. 2. This machine connects to the network through a Broadcom NetXtreme Gigabit Ethernet card. Its purpose is to receive the calls made by User Agent 1 and send back a "180 Ringing" message which is realized with the use of SIPp 3.0 in server mode. The OS of this PC is openSuSE Linux 11, kernel version 2.6.25.16-0.1 with gcc version 4.3.

Two different 1024 bit RSA digital certificates were issued for the Home Proxy and the Inbound Proxy to be used from PrivaSIP-1 and PrivaSIP-2, and the corresponding public keys have been transferred to the UAs. All cryptographic operations are executed by employing the open source OpenSSL library version 0.9.8 g [14]. The measurements where conducted on the network architecture shown in Fig. 3. UA 1 and Home SIP Proxy reside in the same 100 Mbps LAN, while Inbound SIP Proxy and UA 2 reside in another 100 Mbps LAN. The two subnetworks

Table 1
Testbed components.

Machine	CPU	RAM	OS	Software
Low-end UA	500 MHz (AMD mobile Athlon)	256 MB	SuSE Linux 10.0, kernel v. 2.6.13-15	GCC 4.0.2, Twinkle 1.1, OpenSSL 0.9.8 g
High-end UA	2.6 GHz (Intel Pentium 4 hyperthreading)	512 MB	SuSE Linux 10.0, kernel v. 2.6.13-15	GCC 4.0.2, Twinkle 1.1, OpenSSL 0.9.8 g
UA 1	Dual-core 3 GHz (Intel Pentium 4)	1024 MB	OpenSuSE Linux 10.3, kernel v. 2.6.22.18-0.2	GCC 4.2.1, SIPp 3.0, OpenSSL 0.9.8 g
Home SIP Proxy	Dual-core 1.9 GHz (AMD Athlon X2 64)	2048 MB	OpenSuSE Linux 11 (32-bit), kernel v. 2.6.25.16-0.1	GCC 4.3, SER 0.9.6, MySQL 5.0.45-community, OpenSSL 0.9.8 g
Inbound SIP proxy	Dual-core 2.8 GHz (Intel Pentium 4)	1024 MB	OpenSuSE Linux 11, kernel v. 2.6.25.16-0.1	GCC 4.3, SER 0.9.6, OpenSSL 0.9.8 g
UA 2	Dual-core 2.6 GHz (Intel Pentium 4)	512 MB	OpenSuSE Linux 11, kernel v. 2.6.25.16-0.1	GCC 4.3, SIPp 3.0, OpenSSL 0.9.8 g

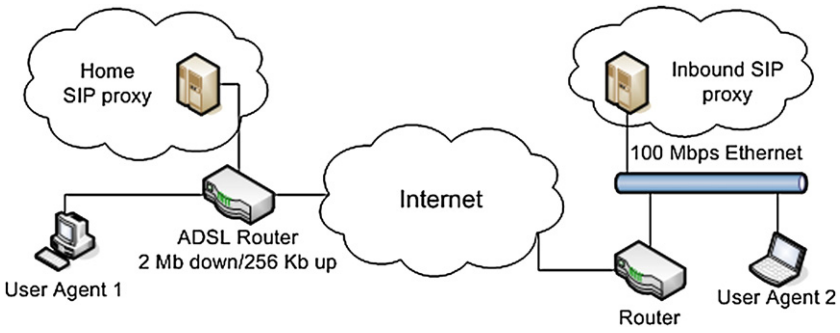

Fig. 3. Testbed network architecture.

connect through the Internet over a 2 Mbit ADSL connection with 2048 Mbps maximum downlink and 256 Kbps maximum uplink speed. The average ping time between the two subnetworks is 22 ms, but this value can only be considered as an indication.

We have made the following modifications to the initial versions of the open source 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. For our first scheme it encrypts only the <From> field while for the second one it encrypts both <From> and <To> fields.
- 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(s). It also decrypts the Digest authentication 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.

For the client delay scenario we have taken measurements with six 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 standard SIP
2. High-end UA with PrivaSIP-1
3. High-end UA with PrivaSIP-2
4. Low-end UA with standard SIP
5. Low-end UA with PrivaSIP-1
6. Low-end UA with PrivaSIP-2.

The measurements for configurations 1, 2, 4 and 5 are those calculated in [6,20]; results for configurations 3 and 6 correspond to our second scheme [20]. Table 2 shows the results for each of the 6 different configurations. Apart from the mean delay, we have included in the table the minimum and maximum delays, the standard deviation of the taken measurements and the 95% confidence interval.

Table 2
SIP and PrivaSIP request preparation delay.

Configuration	Delay (ms)			Standard deviation	Confidence interval (95%)
	Mean	Min	Max		
1	0.16	0.14	1.34	0.07	(0.15, 0.16)
2	0.61	0.55	3.01	0.13	(0.6, 0.62)
3	0.99	0.89	3.29	0.24	(0.97, 1)
4	0.38	0.31	6.11	0.20	(0.37, 0.4)
5	1.6	1.36	8.14	0.26	(1.59, 1.61)
6	2.66	2.33	10.36	0.48	(2.63, 2.69)

The observation of the table reveals that when our schemes are in use the INVITE preparation delay is almost 4 times higher for PrivaSIP-1and 6 to 7 for PrivaSIP-2 compared to standard SIP. This is obviously due to cryptographic operations involved. However, all delays measured are in ms with a maximum of 10.36 ms, meaning that actually 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. This observation is further supported by the calculated confidence intervals.

Fig. 4 shows the impact of hardware configuration on INVITE request preparation delay for each scheme. Here we depict the mean preparation delay values presented in Table 2 adding the corresponding confidence intervals as error bars on the graph. The X axis represents the scheme used, while Y axis shows the INVITE preparation delay in ms.

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 1000 authenticated calls. For each different scheme, server's queue is populated with similar requests, i.e., standard SIP messages for measuring standard SIP's response delays, PrivaSIP-1 messages for our first scheme and PrivaSIP-2 messages for our second scheme. Server's queue population was realized with the SIPp tool, which can create multiple calls with automatically adjusted call rate, so as to keep server's queue at a predefined stable length.

Tables 3–5 show the results for the second scenario. These Tables demonstrate the mean server response delays from the moment the user initiates a call until he gets back a “180 Ringing” message; for each scheme we also include the standard deviation of each mean value and the 95% confidence interval. From these results we infer that there is an overhead in our proposals in comparison to standard

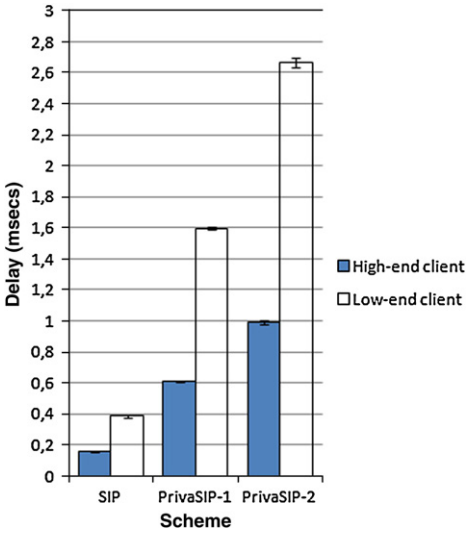

Fig. 4. Mean INVITE preparation delays for different hardware configurations.

Table 3
Mean server response delays for SIP.

Server queue size (calls)	Mean delay (ms)	Standard deviation	Confidence interval (95%)
100	483.2	757.93	(441.02, 525.39)
200	601.72	1019.41	(543.92, 659.52)
300	693.21	1183.6	(623.74, 762.69)
400	738.24	1243.67	(665.66, 810.81)
500	773.7	1306.31	(696.61, 850.79)
600	919.89	1464.04	(832.61, 1007.16)
700	861.13	1364.84	(779.42, 942.83)
800	934.92	1454.72	(847.26, 1022.58)
900	873.53	1361.27	(791.31, 955.75)
1000	1049.62	1461.18	(959.55, 1139.69)

Table 5
Mean server response delays for PrivaSIP-2.

Server queue size (calls)	Mean delay (ms)	Standard deviation	Confidence interval (95%)
100	1244.53	1254.74	(1152.69, 1336.37)
200	1522.85	1592.64	(1382.97, 1662.73)
300	1651.57	1662.91	(1497.93, 1805.21)
400	1634.69	1644.03	(1477.84, 1791.55)
500	1741.45	1744	(1578.31, 1904.59)
600	1576.77	1611.76	(1417.22, 1736.32)
700	1721.93	1701.06	(1562.63, 1881.24)
800	1800.18	1853.96	(1627.34, 1973.02)
900	1858.92	1821.25	(1685.77, 2032.07)
1000	1749.49	1718.94	(1587.58, 1911.39)

SIP regarding the response delays. 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 offered with some additional cost) the requests will be mixed at all SIP proxies involved and the performance penalty will be decreased. Furthermore, as we have already explained, here we consider a worst case scenario regarding the number of cryptographic operations; keeping state information in some SIP Proxies and reusing encrypted URIs will improve the performance of our schemes.

Taking PrivaSIP-2 as an example, in a full roundtrip as shown in Fig. 2, 6 decryptions take place; 4 in Home Proxy (first INVITE's <From> decryption, second INVITE's <From> and Digest username decryption, 180 Ringing <From> decryption) and 2 in Inbound Proxy (INVITE's <To> decryption, 180 Ringing <To> decryption). These decryptions could be limited to 2 if: (a) the client uses the same encrypted URI for all messages of a session, (b) the server stores a correspondence of the encrypted URI and its decrypted value, and (c) Digest username is the same with <From> user ID. To show the performance improvement that can be achieved we take the delays for server queue sizes of 1000 calls. The difference between PrivaSIP-2 and standard SIP, that is 1749.49–1049.62 = 699.87 ms, is mainly due to cryptographic operations. So for each cryptographic operation we have a mean delay of 699.87/6 = 116.65 ms. Following the above optimizations we will have 2 cryptographic operations adding to the delay of standard SIP, i.e., $1049.62 + 2 \times 116.65 = 1282.92$ ms which is a lot better than 1749.49 ms that we measured without any optimization. Of course this is not an accurate value but an estimation, which however shows how much faster our methods can be. It is up to the system administrator to decide and make the proper tradeoff between speed and storage needed for keeping state information.

Fig. 5 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 ms. In each point we have also included the corresponding confidence interval as error bars.

Table 4
Mean server response delays for PrivaSIP-1.

Server queue size (calls)	Mean delay (ms)	Standard deviation	Confidence interval (95%)
100	755.77	992.59	(699.13, 812.4)
200	1030.39	1392.59	(941.46, 1119.32)
300	1145.02	1472.4	(1047.01, 1243.03)
400	1205.5	1536.38	(1100.34, 1310.65)
500	1149.63	1434.47	(1051.21, 1248.05)
600	1155.68	1460.5	(1055.96, 1255.4)
700	1213.87	1543.15	(1108.12, 1319.62)
800	1177.49	1515.59	(1072.72, 1282.25)
900	1279.31	1629.86	(1163.13, 1395.49)
1000	1209.43	1514.79	(1106.75, 1312.11)

4. Privacy level

Before describing related work on SIP ID privacy and comparing it with the proposed schemes we would like to define different levels of ID privacy. The distinction is based on who has access to the real ID of either the caller or the callee or both. We define these privacy levels based on a number of criteria which are shown below in order of importance:

1. The Domains and the callee are considered more trustworthy than other third parties.
2. All Domains engaged are considered more trustworthy than the callee.
3. The Home Domain is considered more trustworthy than other Domains.
4. The ID of some user must be available to as less entities (other than himself) as possible.

The resulting privacy levels are the following starting from no privacy at all:

Level 1: the ID of some user is available to everyone.
Level 2: the ID of some user is available to himself, the user at the other end of the call and all the Proxies of all domains in the call path.
Level 3: the ID of some user is available to himself, the user at the other end of the call and their Home Domains.
Level 4: the ID of some user is available to himself, his Home Domain and the user at the other end of the call.
Level 5: the ID of some user is available to himself and the user at the other end of the call.
Level 6: the ID of some user is only available to himself and his Home Domain.
Level 7: only the owner of the ID has access to it.

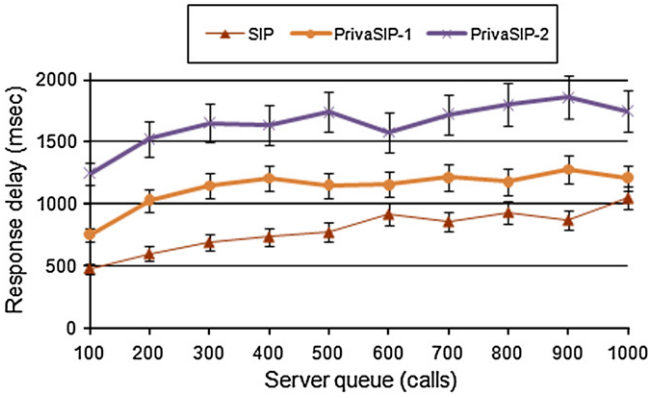

Fig. 5. Comparison of mean server response delays: standard SIP, PrivaSIP-1 and PrivaSIP-2.

5. Related work

The issue of privacy protection is not completely ignored in SIP and this is proved by the fact that [1] includes certain mechanisms that can assist a user in protecting his privacy. These mechanisms can be separated to cryptography based ones which are S/MIME [15], SIPS URI/TLS and IPsec, and the non cryptographic solution of "Anonymous" URI. A different approach is the extension of the basic SIP protocol which led to the solution presented in [16] which will be referred here as "Privacy Mechanism for SIP". This is in fact a general purpose privacy mechanism which has also been used in [17] adapted to the specific needs arose there. In the following these solutions are presented in more detail, while more focus is given on how each solution can be utilized to protect end users' IDs.

5.1. S/MIME

SIP messages consist of two parts: the header and the body. The body part is nothing more than a MIME body, so an obvious solution to protect it is by using the standard way which is S/MIME. Although this may seem out of scope, given that our work focuses on protecting specific SIP headers, S/MIME in the context of SIP can be used to cryptographically protect SIP headers.

S/MIME protects the confidentiality of SIP headers and bodies using digital certificates. In order to protect the privacy of end users S/MIME can encapsulate SIP messages into MIME bodies and encrypt them properly. In our case the encapsulated message can contain the real ID of the caller, while the "outer" message contains a <From> header of the form: "sip:anonymous@anonymizer.invalid". When the called party receives the message, he decrypts the body to find the ID of the caller. What must be noted here is that the ID of the callee cannot be anonymized using the same mechanism since the intermediate SIP Proxies do not have access to the plain MIME body and an anonymous <To> field would made them unable to route the message to the intended recipient.

Although S/MIME can seem as a promising solution there are some obvious weaknesses. First of all the receiver of messages must somehow be aware of the identity of the sender a priori, in order to retrieve the appropriate certificate to decrypt the message body. Another privacy weakness is that the receiver knows the ID of the sender, while the receiver's ID is not protected from third parties. Finally, there is no way to hide the IP addresses of the communicating parties.

5.2. SIPS URI/TLS

It is possible for end users to request that their messages along the whole path to their destination are transported with the use of TLS protocol in order to ensure their privacy protection. This is accomplished with the use of "sips:" instead of "sip:" in a typical SIP URI. While a SIP message having a <To> header of the form: "sip:obrien@miniluv.org" will be visible by anyone, its security enhanced equivalent "sips:obrien@miniluv.org" will request all intermediaries to use TLS in a hop-by-hop manner until the specified domain is reached. After that, the message is handled according to the local security and routing policy.

This approach also presents some worth noting issues. If SIPS URI scheme is selected, then the use of TLS implies the use of TCP as a transport means, while the preferred transport protocol for SIP is UDP. While there is also the solution of DTLS [18], which is the equivalent of TLS using UDP as transport mechanism, it is a scheme that was proposed later than SIP so it is not included in [1]. The main drawback of SIPS URI however is that there is no guaranteed end-to-end protection. While TLS can be used in each hop-by-hop connection, it is not possible to dictate or even be informed somehow that it will be used in every intermediate connection. This can result in two possible attacks; the first one is a downgrade attack, where some intermediate proxy just does not use TLS or replaces "sips:" scheme with "sip:". In the second attack the caller uses plain "sip:" scheme and some intermediate proxy modifies it to a SIPS URI so that the recipient of the message believes that their communication is TLS protected.

5.3. IPsec

For the purposes of SIP, IPsec can be used in a hop-by-hop fashion protecting the data transmitted between two hosts at the network level. The main difference between IPsec and SIPS URI/TLS in the context of SIP is the transparency offered by IPsec to SIP UAs. As it is stated in [1], IPsec will be more suitable in cases where the communicating hosts have already established a trust relationship with one another as opposed to SIPS URI scheme.

What holds for end-to-end protection in SIPS URI also applies here; it is not guaranteed. This is because there is neither an available mechanism to impose the use of IPsec in all intermediate hosts, nor a way for communicating parties to be aware of whether this actually happened or not.

5.4. Anonymous URI

Another approach proposed in [1] for the protection of caller's ID is the use of an Anonymous URI in the <From> field. This URI has meaningless values and it is of the form: "sip:anonymous@anonymous.invalid". It must be stressed here that this Anonymous URI is inserted into the <From> field by the UA itself which means that the SIP Proxy can never have access to the real URI.

The drawback of this solution is that it cannot support UA authentication since no ID is transmitted. A possible workaround could be a UA device shared among many end users. This device will own a specific pair of username and password for authentication purposes which will be the same for all users; however such a solution creates other important security issues like repudiation of actions.

5.5. Privacy mechanism for SIP

The scheme described in [16] is an extension of the basic SIP protocol and defines two ways for the protection of end user's privacy: user and network provided privacy. The end user can choose between these two or utilize both at the same time. When the UA chooses user provided privacy, it populates certain SIP headers with meaningless values, for example <From> field with an Anonymous URI. When network provided privacy is selected an intermediate node is assigned a new logical role for offering anonymization services to UAs, while at the same time is responsible for directing messages from and to the anonymous user as a normal SIP Proxy. In order to enable UAs to request such services a new SIP header is introduced, namely "Privacy-hdr", which takes the following values: header, session, user, none and critical. With the use of one or more of these values the users can ask the network to: obscure headers that cannot be altered without the assistance of an intermediate, for example <Via> and <Contact>, provide anonymization services for the session initiated by the message, cancel any default privacy preferences or mark the criticality of the request for privacy. The recommended way for the UA to communicate with the privacy service provider is by using network or transport layer security protocols.

This mechanism has also been adapted to fit certain requirements in [17]. In this version the user sends a SIP message through a trusted set of Proxies revealing his true ID. When the message is about to leave this trusted domain, the last Proxy withholds the true ID of the user. Similarly to the initial scheme the last Proxy must keep state information in order to route back the responses.

A shortcoming of this method is that the node offering privacy services must keep a significant amount of state information in order to complete the proper routing of the messages. Another issue is that this node can potentially be a single point of failure if replication is not

used. When user provided privacy alone is chosen then what applies for the "Anonymous URI" solution also applies here. The authors of this method have chosen not to consider any privacy considerations arose by the use of authentication mechanisms like Digest authentication. However, a username used in such a method could possibly reveal private information about the end user.

6. Comparison

In this section we will compare our schemes with the related solutions we presented above. First we will analyze the criteria we use for this comparison and then we will show how each scheme responds to these criteria. Finally, a table of comparison will be provided summarizing all the information from the analysis that follows.

6.1. Criteria of comparison

This section lists the criteria used for the comparison of all privacy preserving solutions for SIP. These criteria are:

6.1.1. Cryptography

By this criterion it is examined the use of cryptography for the purposes of each solution. Some schemes are based on cryptography to keep personal information private while others employ other means. A direct implication is that schemes that do not use any kind of cryptography will probably be faster and have less administrative requirements, mainly due to lack of key management.

6.1.2. Authentication

Here we examine whether each solution can support authentication without revealing any private data to non intended parties. More specifically we check if the standard authentication mechanism in SIP, which is Digest authentication, can be utilized without making the real ID of the end user available to third parties.

6.1.3. Public Key Infrastructure (PKI)

With this criterion we separate the proposed solutions based on their PKI requirements. As we will see some of them require a full PKI, others a limited PKI while others no PKI at all.

6.1.4. Anonymity vs. pseudonymity

This criterion indicates what kind of ID is used in the place of the real user ID. This can be a static string like "anonymous@anonymous.invalid" or a completely random string in which case we have a completely anonymous scheme. On the other hand, when the replacement ID is produced in some way from the real ID we have a scheme based on pseudonymity. The most notable difference here is that the person receiving a call from a UA using a pseudonym can always return the call using this pseudonym something that is not possible with anonymous schemes. Also, in a poor designed scheme that uses pseudonyms, a user can be tracked down when, for example, is using the same pseudonym repeatedly, even if the correspondence between the real ID and the user ID is kept secret.

6.1.5. Inter-domain agreements

One of the most common preconditions in schemes offering security services in multidomain environments is that different administrative domains must have pre-existing trust agreements between them. This limits the number of users' choices only to networks that belong to co-operative domains. In our comparison we examine whether each solution needs such pre-existing agreements between domains in order to offer ID privacy to end users.

6.1.6. Multidomain support

Here we examine whether a solution can support its privacy features when operating in an environment composed of different administrative domains. These domains can belong to different operators and/or service providers. The difference between "Multi-domain support" and "Inter-Domain agreements" is that a scheme can support multidomain environments without requiring pre-arranged inter-domain agreements; when a solution requires inter-domain agreements, obviously supports multidomain environments. A scheme can either fully support multidomain environments or not.

6.1.7. Untrusted proxies

When a UA initiates a multimedia session its request can travel through untrusted SIP Proxies until it reaches its Home Proxy which is considered trusted. Our purpose here is to check whether each solution can guarantee UA's privacy protection even when SIP messages traverse through untrusted proxies.

6.1.8. Domain name protection

Since we focus on schemes that preserve the privacy of end users we are concerned on protecting as much private information as possible. With this criterion we examine if each method protects among other things the Home Domain's name of each or both the communicating UAs. While domain name is private information its protection is not considered of ultimate importance since its disclosure does not directly reveal the ID of the end user.

6.1.9. IP address protection

What holds for domain names also holds for protecting each end user's IP address. It is private information which is not considered crucial and cannot directly lead to the real ID of the user. However, under some circumstances, it can reveal the current position of the user and in extreme situations, combined with other personal information, even his real ID.

6.1.10. Privacy level

This criterion shows in which of the privacy levels listed in Section 4 each method is classified. The classification is based on "how much" privacy each method offers; thus the higher the level, the higher the privacy offered by each method. In order to be more practical, numbers 1 to 7 will be used to indicate which of these levels is reached.

6.1.11. Hop-by-hop vs. end-to-end privacy

As we have already seen the establishment of a SIP session typically includes a number of intermediate nodes. With this criterion we check whether each method can guarantee users' ID privacy in a hop-by-hop or an end-to-end manner; obviously the second is the preferred one since only this way we can be sure that privacy was not compromised along the session path.

6.1.12. Stateful vs. stateless mode

Here we examine whether each scheme requires SIP Proxies to be stateful or stateless in order to be fully operational. Stateful proxies keep state information for each ongoing session something that speeds up or make possible the offer of specific services, however leads to a need for more storage resources. Stateless proxies on the other hand do not store any information regarding sessions so they have less storage needs and have higher response delays. While each mode has its own advantages over the other, in some occasions some services may be able to run only in one of the two.

6.1.13. Deployment

This criterion indicates the easiness of deployment of a scheme. We will use a qualitative measurement based on empirical observation. We define three degrees of ease of deployment: easy, medium and difficult.

6.2. Schemes analysis

Here we will further comment on each scheme based on the thirteen aforementioned criteria.

6.2.1. S/MIME

6.2.1.1. Cryptography. S/MIME cryptographically protects various SIP headers using public key cryptography and digital certificates of end users.

6.2.1.2. Authentication. In [1] it is mentioned that encrypting <Authorization> and <WWW-Authenticate>header fields is not considered useful and any encrypted form of these fields will be ignored. This means that ID privacy during authentication is not supported and anyone can have access to all usernames of end users when they authenticate.

6.2.1.3. PKI. Since S/MIME uses public key cryptography it is straightforward that a sort of PKI is required. In this occasion a full PKI is needed where a digital certificate must be issued for every end user.

6.2.1.4. Anonymity vs. pseudonymity. In this solution a meaningless value is used in the "outer"<From>field, while the real ID is placed into the encrypted MIME body. While the real ID exists in every such message it is encrypted together with other values thus it cannot considered as pseudonym; naturally this solution is based on anonymity.

6.2.1.5. Inter-domain agreements. This scheme does not need any pre-existing agreements between administrative domains. Each user must have some kind of trust agreement with the party he is communicating with.

6.2.1.6. Multidomain support. S/MIME supports multidomain environments since SIP Proxies do not intervene in any way to the part of the message that preserves end user's privacy.

6.2.1.7. Untrusted proxies. This solution protects user's ID even when the relevant SIP messages travel through untrusted proxies. However, as already mentioned above, it cannot protect the username used for Digest authentication; thus we consider S/MIME as a method that is not supporting privacy through untrusted proxies.

6.2.1.8. Domain name protection. The Home Domain name of the caller is not explicitly revealed, however an eavesdropper can discover which domains communicate with each other. On the other hand the Home Domain name of the callee is not protected.

6.2.1.9. IP address protection. The IP addresses of the communicating parties are not protected.

6.2.1.10. Privacy level. This scheme reaches Level 5 concerning caller's ID since caller's real ID is available only to the caller and the callee. Regarding callee's ID S/MIME offers no protection so it reaches Level 1.

6.2.1.11. Hop-by-hop vs. end-to-end privacy. The privacy protection of this solution is offered in an end-to-end fashion.

6.2.1.12. Stateful vs. stateless mode. SIP proxies do not play any active role in privacy protection in this scheme so both modes are supported.

6.2.1.13. Deployment. The utilization of this solution mandates the deployment of a full PKI; as every typical PKI this includes a number of administrative actions like issuing digital certificates to all end users and revoking them when this is necessary. Another issue with S/MIME is that the callee must know a priori which the caller is in order to be able to choose and acquire the right public key certificate. For those reasons this scheme is considered to have difficult deployment.

6.2.2. SIPS URI/TLS

6.2.2.1. Cryptography. SIPS/URI utilizes TLS to protect TCP sessions between SIP network elements; obviously cryptography is part of this solution.

6.2.2.2. Authentication. Digest authentication is supported as is by this solution and the usernames are protected as well.

6.2.2.3. PKI. A full PKI is needed since TLS is used. According to this scheme digital certificates for all communicating users and intermediate SIP servers must be issued. Since a PKI is a requirement certificate acquisition, management and revocation is also an issue here.

6.2.2.4. Anonymity vs. pseudonymity. SIP messages are transmitted through secure channels, therefore, no user ID is revealed; this means that this solution retains user's anonymity.

6.2.2.5. Inter-domain agreements. This scheme requires pre-existing agreements between administrative domains so that SIP Proxies belonging to different domains can establish a secure channel with the use of TLS. These agreements can be indirect based on digital certificates, i.e., cross-certifications, and an existing PKI. It must be noted here that it is not obligatory for communicating users to have explicit trust agreements between them.

6.2.2.6. Multidomain support. SIPS/URI supports multidomain environments which have some sort of trust agreements between them, e.g., have been cross-certified beforehand, as already stated above.

6.2.2.7. Untrusted proxies. This solution should not be used when untrusted SIP Proxies exist in the communication path. If this is the case then it is possible that these untrusted Proxies will not use TLS, so no protection is offered to the communicating parties at all.

6.2.2.8. Domain name protection. When SIPS/URI is used the domain names of each of the communicating parties is protected from eavesdroppers without however being hidden from intermediate Proxies. There are also some cases where everyone can have access to this information like, for example, when only two domains intervene between the two communicating parties so that it is obvious who belongs to which domain.

6.2.2.9. IP address protection. The IP addresses of the communicating parties are not protected.

6.2.2.10. Privacy level. For both caller's and callee's IDs the solution of SIPS URI reaches Level 2 since both real IDs are available to all SIP Proxies in the call path.

6.2.2.11. Hop-by-hop vs. end-to-end privacy. The privacy protection of this solution is offered in a hop-by-hop fashion.

6.2.2.12. Stateful vs. stateless mode. Since TLS is used, we need a server that is stateful at the transport level which means that storage requirements are higher. At the application level where SIP operates there is no special need to keep state information. Based on these two observations and taking the SIP Proxy machine as a whole we can argue that it operates in stateful mode.

6.2.2.13. Deployment. The utilization of this solution has as prerequisite the deployment of a full PKI which issues digital certificates to all end users and intermediate SIP Proxies. Also, currently, there are

few SIP clients and network servers that implement TLS and SIPS respectively. Taking into account the administrative effort required to setup a full PKI and the changes needed in the existing infrastructure, this scheme is considered to have difficult deployment.

6.2.3. IPsec

6.2.3.1. Cryptography. IPsec is based on cryptography to protect data exchanged between two communicating parties.

6.2.3.2. Authentication. Digest authentication is supported and the corresponding authentication usernames are protected by IPsec.

6.2.3.3. PKI. IPsec usually bases its operation in pre-shared secret values so no PKI is required. However, if IKE [19] is used with certificates then the deployment of a PKI is necessary.

6.2.3.4. Anonymity vs. pseudonymity. SIP messages are transmitted through secure channels therefore no user ID is revealed; this means that this solution retains user's anonymity.

6.2.3.5. Inter-domain agreements. This scheme is based on already established trust relationships between the two communicating parties. Therefore, there should be some kind of pre-existing agreement between administrative domains so that Proxies belonging to different domains can establish secure channels with the use of IPsec.

6.2.3.6. Multidomain support. This solution can also be utilized in environments where multiple administrative domains exist.

6.2.3.7. Untrusted proxies. What applies to SIPS URI also applies here.

6.2.3.8. Domain name protection. What applies to SIPS URI applies here as well.

6.2.3.9. IP address protection. What applies to SIPS URI also applies here.

6.2.3.10. Privacy level. What applies to SIPS URI applies as well.

6.2.3.11. Hop-by-hop vs. end-to-end privacy. What applies to SIPS URI also applies here.

6.2.3.12. Stateful vs. stateless mode. When IPsec is used, SIP proxies can operate in either of these two modes.

6.2.3.13. Deployment. The utilization of this solution requires every intermediate node in the call path to have a shared secret with every node it communicates with. This makes it a solution with difficult deployment. The number of IKE pre-configured keys needed in a symmetric key system with n network elements communicating with each other is $O(n^2)$. Also, as already mentioned, if IKE is used with certificates then a full PKI is also required.

6.2.4. Anonymous URI

6.2.4.1. Cryptography. This solution does not utilize any kind of cryptography.

6.2.4.2. Authentication. Anonymous URI can support Digest authentication but this would mean that either the username must be revealed or an “anonymous” username must be used.

6.2.4.3. PKI. No PKI is required for this scheme.

6.2.4.4. Anonymity vs. pseudonymity. Since no caller ID is transmitted this is a solution based on anonymity.

6.2.4.5. Domain agreements. This scheme does not require any pre-existing agreements between administrative domains.

6.2.4.6. Multidomain support. Anonymous URI supports multidomain environments without any modification.

6.2.4.7. Untrusted proxies. Anonymous URI can preserve user's anonymity even when untrusted proxies reside in the path between the caller and the callee.

6.2.4.8. Inter-domain name protection. When Anonymous URI is utilized the domain name of the caller is never transmitted, while anyone has access to the callee's domain name.

6.2.4.9. IP address protection. The IP addresses of the communicating parties are not protected.

6.2.4.10. Privacy level. Regarding caller's ID, Anonymous URI is at Level 7, because only the caller is aware of his own ID, while for callee's ID no protection at all is offered resulting at privacy Level 1.

6.2.4.11. Hop-by-hop vs. end-to-end privacy. This scheme offers end-to-end privacy for caller's ID.

6.2.4.12. Stateful vs. stateless mode. This is a solution that can be supported either by stateful or stateless SIP proxies.

6.2.4.13. Deployment. Anonymous URI is a method with easy deployment since no modification to the existing infrastructure is required.

6.2.5. Privacy mechanism for SIP

This mechanism has two ways for protecting user's privacy: user and network provided privacy. When user provided privacy is employed then what applies for Anonymous URI as analyzed in the previous section, also applies here. The following analysis is valid when network or both user and network provided privacy is used.

6.2.5.1. Cryptography. This scheme does not base its operation on cryptography. However, the recommended way the UA contacts its Home Domain is over a TLS session; thus we consider here that cryptography is part of this solution.

6.2.5.2. Authentication. While digest authentication can be used with this method, the username is not protected at all. In some occasions this can result in privacy violation, for example when the username is the same as the user ID part of SIP URI.

6.2.5.3. PKI. Considering that TLS will be used, a limited PKI is needed for the management of certificates for SIP proxies.

6.2.5.4. Anonymity vs. pseudonymity. This method uses real SIP URIs inside trusted domains while replacing them with Anonymous URIs when SIP messages leave these trusted domains. Thus, it is a method that offers anonymity to its users.

6.2.5.5. Inter-domain agreements. In this scheme a privacy service entity is needed which can be, for example, a trusted SIP Proxy. If this Proxy does not belong to the user's Home Domain then a trust agreement is needed between the Home Domain and Proxy's domain so that the end user can trust the latter.

6.2.5.6. Multidomain support. This method can support multidomain environments but only strictly under the assumption that these domains have established trust agreements with each other. In other

words, if a user is located in a place where there is no administrative domain with trust agreement with his Home Domain then he cannot use the features offered by this solution.

6.2.5.7. Untrusted proxies. This mechanism cannot guarantee the protection of user's privacy when SIP messages are transmitted through untrusted Proxies before reaching his Home Domain.

6.2.5.8. Domain name protection. When SIP messages leave a trusted domain they are anonymized; however, the responses must follow the same path back in order to be routed properly to the sender. Thus, the name of the caller's domain cannot be kept secret.

6.2.5.9. IP address protection. The IP addresses of the communicating parties are not protected.

6.2.5.10. Privacy level. For caller's ID this mechanism reaches Level 6 while for callee's ID it is at Level 1.

6.2.5.11. Hop-by-hop vs. end-to-end privacy. While in [16] it is suggested that TLS should be used from UA to its Home Domain's Proxy, this solution as a whole is considered an end-to-end privacy preserving one. This is because inside the trusted domains we can be sure that TLS or other protection methods will be used while outside the domain no real ID is transmitted.

6.2.5.12. Stateful vs. stateless mode. This mechanism requires state information to be kept in certain Proxies, thus it can only be supported by stateful Proxies.

6.2.5.13. Deployment. Privacy mechanism for SIP is considered a solution which requires medium deployment effort. The UAs and the Proxies must be modified in order to be able to process the new privacy header; in addition to that, Proxies must have the proper logic to withhold user IDs when this is necessary and route responses properly.

6.2.6. PrivaSIP-1

6.2.6.1. Cryptography. This solution protects user's privacy based on cryptography.

6.2.6.2. Authentication. Our first scheme supports Digest authentication; furthermore during the authentication process the username of the caller is protected.

6.2.6.3. PKI. A limited PKI is needed. We use the term “limited” because digital certificates will be issued and managed only for Proxies and not for end users. Moreover, managing certificates for a small number of trusted servers is easier than doing the same for all SIP users.

6.2.6.4. Anonymity vs. pseudonymity. The protection of user's ID involves the encryption of this ID and the transmission of its encrypted form. This encrypted form is a pseudonym and the real ID can be recovered by this pseudonym by entitled entities.

6.2.6.5. Inter-domain agreements. This scheme does not require any kind of trust agreement to exist between different administrative domains.

6.2.6.6. Multidomain support. This method can support multidomain environments even when different administrative domains do not have established any kind of trust agreement between them.

6.2.6.7. Untrusted proxies. This mechanism can protect caller's IDs and Digest authentication passwords even when untrusted proxies exist in the path between the user and his Home Domain.

6.2.6.8. Domain name protection. Our scheme does not protect the name of the caller's Home Domain.

6.2.6.9. IP address protection. The IP addresses of the communicating parties are not protected.

6.2.6.10. Privacy level. For caller's ID our mechanism reaches Level 6 while for callee's ID it is at Level 1.

6.2.6.11. Hop-by-hop vs. end-to-end privacy. Our scheme offers end-to-end privacy.

6.2.6.12. Stateful vs. stateless mode. This mechanism can be supported by either stateful or stateless SIP Proxies.

6.2.6.13. Deployment. The modification needed by our scheme in UAs and Proxies is the addition of encryption/decryption abilities. Apart from this, a PKI is needed which is however limited to manage certificates issued only to Proxies. Due to the limited nature of the PKI we consider our method to require medium deployment effort.

6.2.7. PrivaSIP-2

6.2.7.1. Cryptography. Our second scheme is also based on cryptography.

6.2.7.2. Authentication. This method supports Digest authentication while at the same time protecting the username of the caller.

6.2.7.3. PKI. The same applies here as in PrivaSIP-1; a limited PKI is needed.

6.2.7.4. Anonymity vs. pseudonymity. Both users IDs (caller's and callee's ID) are encrypted prior to their transmission and are pseudonyms of the real IDs.

6.2.7.5. Inter-domain agreements. Similarly to PrivaSIP-1, this scheme does not require any trust agreements between different administrative domains.

6.2.7.6. Multidomain support. Multidomain environments can be supported in this method even when different administrative domains do not have established any kind of trust agreement between them.

6.2.7.7. Untrusted proxies. Our second scheme protects both caller's and callee's IDs and Digest authentication passwords even when untrusted proxies exist anywhere in the call path.

6.2.7.8. Domain name protection. Our scheme does not protect domain names.

6.2.7.9. IP address protection. The IP addresses of the communicating parties are not protected.

6.2.7.10. Privacy level. For caller's ID our mechanism reaches Level 6 while for callee's ID it reaches Level 4.

6.2.7.11. Hop-by-hop vs. end-to-end privacy. Our scheme offers end-to-end privacy.

6.2.7.12. Stateful vs. stateless mode. This mechanism can be supported by either stateful or stateless SIP Proxies.

6.2.7.13. Deployment. What applies in PrivaSIP-1, also applies here; hence our method needs medium deployment effort (Table 6).

7. Discussion

In this section we will comment on some interesting points from the observation of Table 3; the first one has to do with ID hiding. In some occasions it is desirable from the caller not to reveal his ID to the callee. This ID hiding type is supported by our schemes and by other schemes as well; these other schemes are “Anonymous URI” and “Privacy mechanism for SIP”. The difference here is that only the two PrivaSIP schemes can support this feature while at the same time protecting the Digest username during the authentication process.

As the most important advantage of our methods we consider their ability to maintain their privacy protecting features while operating through untrusted domains, even when these domains are placed between the caller and his Home Domain. While S/MIME can also protect the user ID, it cannot protect his username during Digest authentication. Furthermore, it cannot offer caller's ID hiding from the callee.

Another consideration is that only “Anonymous URI” can protect the Home Domain name of the caller; however this method is less practical since it cannot support authentication. Regarding the IP addresses of the communicating parties it is evident that no method can effectively protect them from eavesdroppers. While both domain names and IP addresses are considered private information, they should remain publicly available so that the two parties can communicate with each other during, as well as after, the session establishment.

Our methods have the highest possible privacy level regarding real world practical SIP applications. We define such applications as having the following requirements:

- User authentication (which is required among others for billing purposes)
- The real ID of the user must be available to as less entities as possible.
- Privacy protection must be assured even through untrusted proxies in an end-to-end fashion.

Having these requirements in mind, both our methods have the highest level of privacy together with “Privacy mechanism for SIP” when caller's ID privacy is required. While “Anonymous URI” is at level 7, it does not support user authentication as already mentioned. Regarding callee's ID privacy our second scheme, PrivaSIP-2, has the highest privacy level of all schemes. All other methods are at level 1 or 2 and this shows that callee's ID privacy has not been taken into consideration at all by these methods.

One final remark concerning our schemes is the acquisition of Proxy certificates. Throughout this paper we assume that the UAs have in their possession the digital certificates of the Proxies they need. This is a logical assumption concerning the Home Proxy certificate of each user; however, the same cannot be straightforwardly asserted for other Proxies. Thus, when PrivaSIP-2 is utilized the caller's UA should first acquire and check the certificate of the callee's Home Proxy and then proceed to the protection of the messages. This however happens usually once and stands for multiple sessions, i.e., until the certificate of the corresponding foreign SIP Proxy expires.

8. Contribution

In this section we would like to summarize and clear out the contribution of this paper compared to previous work. In this paper two SIP privacy preserving protocols are presented; the first one, namely PrivaSIP-1, has already been presented in [6,20], while the second one, PrivaSIP-2, has been proposed in [20]. Regarding the testbed experiments, as it has been demonstrated in Section 3, we have both client and server side scenarios. In [6,20] we have measured the SIP INVITE preparation delay for the client and here we reuse the same measurements for standard SIP and PrivaSIP-1; all results presented here for PrivaSIP-2 are from [21]. The same applies to server side measurements. Sections 4 to 7 review related work on SIP privacy preserving methods and provide a comparison with our proposed protocols; this SIP privacy survey introduced in this article is novel and no such review exists to our knowledge.

9. Conclusions

It is envisioned that in the near future SIP will co-exist or even supersede traditional telephony systems like PSTN. Before this becomes reality certain security issues must be solved. While SIP is a simple and easy to deploy protocol, it turns out that some of the security problems related with it are hard to solve. One such problem is privacy since SIP messages cannot be cryptographically protected as a whole.

As we already showed SIP has a number of security and especially privacy protecting mechanisms; however some privacy issues are still open. Here we concentrate on the protection of communicating parties IDs in an easy to deploy manner. We also review existing solutions focusing on how each method can protect user IDs and comparing them with our proposals.

We argued that our methods can protect user IDs more effectively and in cases where existing methods fail to satisfy users' privacy needs. This is especially true when a fair balancing between privacy and performance is terminus. Our quantitative analysis through testbed experimentation showed that for the client side the delay is negligible, while our methods turns out to be quite expensive in terms of time delay

Table 6
Privacy schemes comparison.

Schemes Criteria		S/MIME	SIPS URI/TLS	IPsec	Anonymous URI	Privacy mechanism	PrivaSIP-1	PrivaSIP-2
Cryptography		✓	✓	✓	x	✓	✓	✓
Authentication		×	✓	✓	x	x	✓	✓
PKI		Full	Full	x	x	Limited	Limited	Limited
Anonymity vs. pseudonymity		Anonymity	Anonymity	Anonymity	Anonymity	Anonymity	Pseudonymity	Pseudonymity
Inter-Domain agreements		x	✓	✓	x	✓	x	x
Multidomain support		✓	✓	✓	✓	✓	✓	✓
Untrusted proxies		x	X	x	x	x	✓	✓
Domain name protection		x	x	x	✓	x	x	x
IP address protection		x	x	x	x	x	x	x
Privacy level	Caller	5	2	2	7	6	6	6
	Callee	1	2	2	1	1	1	4
Hop-by-hop vs. end-to-end privacy		End-to-end	Hop-by-hop	Hop-by-hop	End-to-end	End-to-end	End-to-end	End-to-end
Stateful vs. stateless		Both	Stateful	Both	Both	Stateful	Both	Both
Deployment		Difficult	Difficult	Difficult	Easy	Medium	Medium	Medium

✓: supported/required.
×: not supported/not required.

for SIP Proxies. We have already discussed certain possible improvements that could alleviate these delays; in addition our future work will concentrate in finding ways to further improve the performance of PrivaSIP.

Acknowledgements

The authors would like to thank Mrs. Evangelia Papanagiotou for her assistance in statistical calculations.

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 (20% from the Greek Ministry of Development-General Secretariat of Research and Technology and 80% from E.U.-European Social Fund).

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: Session Initiation Protocol, RFC 3261, June 2002.
[2] 3rd Generation Partnership Project (3GPP) Consortium, http://www.3gpp.org.
[3] H. Schulzrinne, E. Wedlund, Application-layer mobility using SIP, Mobile Computing and Communications Review 4 (3) (July 2000) 47–57 ACM SIGMOBILE.
[4] D. Geneiatakis, G. Kambourakis, T. Dagiuklas, C. Lambrinoudakis, S. Gritzalis, SIP security mechanisms: a state-of-the-art review, Proceedings of the Fifth International Network Conference (INC 2005), Samos, Greece, July 2005.
[5] D. Geneiatakis, G. Kambourakis, C. Lambrinoudakis, T. Dagiuklas, S. Gritzalis, A framework for protecting a SIP-based infrastructure against malformed message attacks, Computer Networks, vol. 51, Issue 10, Elsevier, July 2007, pp. 2580–2593.
[6] G. Karopoulos, G. Kambourakis, S. Gritzalis, Caller Identity Privacy in SIP Heterogeneous Realms: A Practical Solution, in: A. Zanella, et al., (Eds.), 3rd Workshop on Multimedia Applications over Wireless Networks (MediaWin 2008), IEEE Computer Society Press, Marakkech, Morocco, July 2008.
[7] J. Rosenberg, C. Jennings, The Session Initiation Protocol (SIP) and Spam, RFC 5039, January 2008.
[8] A. Pfitzmann, M. Hansen, Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management—A Consolidated Proposal for Terminology, Version v0.31, Feb. 15 2008 available at http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.
[9] M. Bellare, P. Rogaway, Optimal Asymmetric Encryption—How to encrypt with RSA, in: A. De Santis (Ed.), Extended abstract in Advances in Cryptology—Eurocrypt '94 Proceedings, LNCS, vol. 950, Springer-Verlag, 1995.
[10] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, HTTP Authentication: Basic and Digest Access Authentication, RFC 2617, June 1999.
[11] SIP Express Router (SER), free, open source SIP server, available at http://www.iptel.org/ser.
[12] Twinkle SIP softphone, open source, available at http://www.twinklephone.com.
[13] SIPp, open source performance testing tool for SIP, available at http://sipp.sourceforge.net.
[14] OpenSSL, open source SSL/TLS library, available at http://www.openssl.org/.
[15] B. Ramsdell, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification, RFC 3851, July 2004.
[16] J. Peterson, A Privacy Mechanism for the Session Initiation Protocol (SIP), RFC 3323, November 2002.
[17] C. Jennings, J. Peterson, M. Watson, Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, RFC 3325, November 2002.
[18] E. Rescorla, N. Modadugu, Datagram Transport Layer Security, RFC 4347, April 2006.
[19] C. Kaufman, Internet Key Exchange (IKEv2) Protocol, RFC 4306, December 2005.
[20] G. Karopoulos, G. Kambourakis, S. Gritzalis, E. Konstantinou, "A Framework for Identity Privacy in SIP", Journal of Network and Computer Applications, vol. 33, Issue 1, Elsevier, 2010, pp. 16–28, ISSN: 1084-8045.
[21] Min-Xiou Chen, Chen-Jui Peng, Ren-Hung Hwang, SSIP: Split a SIP Session over Multiple Devices, Computer Standards & Interfaces 29 (5) (July 2007) 531–545.
[22] Liufei Wu, Yuqing Zhang, Fengjiao Wang, A new provably secure authentication and key agreement protocol for SIP using ECC, Computer Standards & Interfaces 31 (2) (February 2009) 286–291.

Giorgos KAROPOULOS (gkar@aegean.gr) is currently a Postdoctoral research fellow at the Info-Sec-Lab of the Department of Information and Communication Systems Engineering, University of the Aegean. He holds a diploma in Information and Communication Systems Engineering, a MSc in Information and Communication Systems Security, and a PhD in Computer Network Security from the University of the Aegean. His current research focus is in mobile multimedia security in all-IP heterogeneous networks.

Georgios KAMBOURAKIS received a Diploma in Applied Informatics from Athens University of Economics and Business in 1993 and a Ph.D. in Information and Communication Systems Engineering from the Department of Information and Communications Systems Engineering of the University of Aegean. He also holds a M.Ed. from the Hellenic Open University. Currently, Dr. Kambourakis is a Lecturer at the Department of Information and Communication Systems Engineering of the University of the Aegean, Greece. His main research interests are in the fields of mobile and wireless networks security and privacy, VoIP security and mLearning. He has been involved in several national and EU funded R&D projects in the areas of Information and Communication Systems Security. He is a reviewer of several IEEE and other international journals and has served as a technical program committee member in numerous conferences.

Prof. Dr. Stefanos GRITZALIS holds a BSc in Physics, a MSc in Electronic Automation, and a PhD in Informatics all from the University of Athens, Greece. Currently he is the Deputy Head of the Department of Information and Communication Systems Engineering, University of the Aegean, Greece and the Director of the Laboratory of Information and Communication Systems Security (Info-Sec-Lab). He has been involved in several national and EU funded R&D ICT projects. His published scientific work includes several books on Information and Communication Technologies topics, and more than 200 journal and national and international conference papers. The focus of these publications is on Information and Communications Security and Privacy. He has led more than 30 international conferences and workshops as General Chair or Program Committee Chair, and has served on more than 200 Program Committees of international conferences and workshops. He was an elected Member of the Board (Secretary General, Treasurer) of the Greek Computer Society.