A Cost-Effective Mechanism for Protecting SIP Based Internet Telephony Services Against Signaling Attacks

Dimitris Geneiatakis and Costas Lambrinoudakis
Laboratory of Information and Communication Systems Security
Department of Information and Communications Systems Engineering, University of the Aegean
Karlovassi, GR-83200 Samos, Greece
Tel:+30-22730-82247
{dgen,clam}@aegean.gr

ABSTRACT
Internet Telephony services offer several new business opportunities to telecommunication providers. However, they also introduce several security flaws that can be exploited by various attacks, thus raising the need for the employment of suitable security measures during the provision of the service. Signaling attacks, a type of Denial of Service attacks, are an indicative example. In this paper we present a cost-effective mechanism, namely the *Integrity-Auth* mechanism, for protecting SIP-based telephony services from signaling attacks. The focus is on the evaluation of the mechanism in terms of the processing overheads that it introduces in various different scenarios.

Keywords
Session Initiation Protocol, Signaling Attacks, Security

1. INTRODUCTION
Applications and services employed in open architectures, like the Internet, do not only inherit the benefits resulting from the characteristics of such architectures, but also the new security threats that are inevitably raised. It is well known that Internet's protocols are susceptible to a plethora of vulnerabilities and attacks [1][2]. Furthermore, attackers keep on searching for new security flaws of the Internet protocol stack. Consequently, Internet should be considered as a hostile environment, at least by any critical real-time application like *Internet Telephony*.

Today there are various researchers that have identified security flaws in Internet Telephony Services (ITS) [3]-[6], mainly focusing on those utilizing the Session Initiation Protocol (SIP) [7] as their signaling protocol. The SIP "preference" originates from the fact that SIP has been adopted by various standardization organizations as the predominant protocol for both wireline and wireless world in the Next Generation Networks (NGN) era. One of the most important vulnerabilities is the fact that a malicious user may illegally modify selected session parameters and thus launch a signaling attack [3]. In such cases the malicious user sends a carefully crafted (well formed) SIP message in order to either cause a Denial of Service (by illegally terminating a session, by modifying the voice parameters etc.), or to gain unauthorized access to the provided service.

It is therefore necessary for SIP realms to employ the appropriate security mechanism in order to protect ITSs against signaling attacks. Currently, there are very few attempts, published in the literature, addressing this issue [8][9]. Particularly, [8] presents a mechanism that could be utilized against the BYE signaling attack. The main idea of this mechanism is based on cross protocol intrusion detection, recognizing a BYE signaling attack if voice data (RTP messages) follow the terminating signaling message BYE, during some specific time window. The main drawback of that solution is that a malicious user can bypass the aforementioned detection mechanism if before sending the BYE message to the other participant he causes a DoS situation. Furthermore, a malicious user can generate false alarms by sending voice data after the legitimate user has sent the terminating SIP BYE message. Finally, this scheme does not protect end users against man-in-the-middle signaling attacks. On the other hand, [9] presents a pro-active security mechanism that protects SIP's session responses against illegal modifications and guarantees their authenticity and integrity. This means that the latter solution is effective in cases of man-in-the-middle attacks where the malicious user acts as a proxy generating spoofed responses, offering security services to only one of the participants (the caller) of a multimedia session and not to the other (the callee). As a result the specific mechanism cannot be utilized to provide protection against all types of signaling attacks. Thus, more advanced mechanisms are necessary in order to improve the security level of SIP based internet telephony services.

In this paper we present and evaluate, through different scenarios, a practical and cost-effective approach, namely the *Integrity-Auth* mechanism [10], that can be applied in SIP based services for protecting them against signaling attacks. The rest of paper is structured as follows: Section 2 briefly describes the signaling attacks that can be launched against SIP based services. Section 3 presents the proposed protection mechanism, while section 4 evaluates the *Integrity-Auth* scheme in terms of the overhead introduced at the SIP server side, as well as at the end-user terminal. Section 5 concludes the paper.

2. SIGNALING ATTACKS
A signaling attack could be defined as any attempt to illegally modify the specific session of a state (i.e. terminating, canceling, updating) in order to either cause a DoS or to gain unauthorized access to the provided service. Signaling attacks exploit the fact that there are no appropriate authentication and integrity check

Mobimedia 2008 July 7-9, 2008, Oulu, Finland.
 ISBN 978-963-9799-25-7
DOI 10.4108/ICST.MOBIMEDIA2008.3857

mechanisms [3] developed for SIP realms. Furthermore, the easy access to the IP-based networks gives the chance to malicious users to "discover" the parameters required for launching such an attack. Specifically, consider the case where a legitimate user initiates a call by sending the appropriate SIP INVITE message. The malicious user, who wishes to cancel this specific session, should first identify the corresponding parameters of the initial SIP INVITE message, by utilizing sniffing tools like Wireshark (http://www.wireshark.org/). Following that, at any future time the malicious user can craft the appropriate SIP CANCEL message in order to terminate the pending session, without the legal user realizing the reason for which his request has been canceled. The steps followed during this type of attack are depicted in Figure 1. More details about signaling attacks can be found in [3].

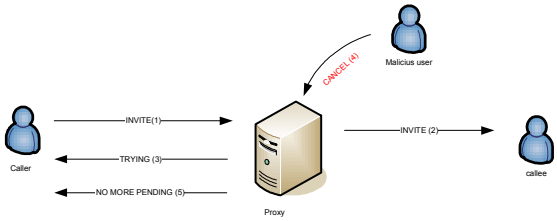

Figure 1. An Example of a SIP Signaling Attack

3. THE INTEGRITY-AUTH SCHEME

3.1 General Description

As already mentioned, today there is no appropriate mechanism for validating the authenticity and integrity of a SIP message. To this direction, [10] has suggested the introduction of the *Integrity-Auth* scheme. The *Integrity-Auth,* as its name implies, offers integrity and authenticity services, shielding SIP based ITS against signaling attacks. This is achieved through the introduction of a new header named Integrity-Auth, which includes the appropriate credentials for validating both message integrity and authenticity. The grammar of the proposed Integrity-Auth header is presented in Figure 2, while the corresponding credential value is computed through the following formula:

Head.-Val=Hash(SIP_Message|Random,Hash(PWD|Random) (1)

Note that only legally authorized users can generate the credential values that correspond to a specific SIP message. Besides, the malicious user is unable to find the user's password from the integrity-auth value, as hash functions are irreversible. However, if the user's password has been revealed, then the security of the entire scheme is sacrificed, as is the case in any password-based system.

```
Integrity-Auth="Integrity-Auth" HCOLON integrity-auth-value
integrity-auth-value= credentials-value;algrotithm;nonce
algorithm="algorithm" EQUAL alg-value
alg-value="MD5|SHA1"
credentials-value=quoted-string
```

Figure 2. Integrity-Auth Header Grammar

Whenever a SIP entity, which employs the *Integrity-Auth* scheme, receives a SIP message (e.g a request), it extracts the initial value of the *Integrity-Auth* header and checks its validity by re-computing the *Integrity Auth* value according to formula (1); if the SIP entity is a SIP UA terminal the password is provided by the user, otherwise it should be searched at the corresponding database. If the two values match the processing of the message continues, otherwise the received message is discarded and an alarm for a signaling attack is triggered. Figure 3 provides the message flow for validating the genuineness of a SIP message.

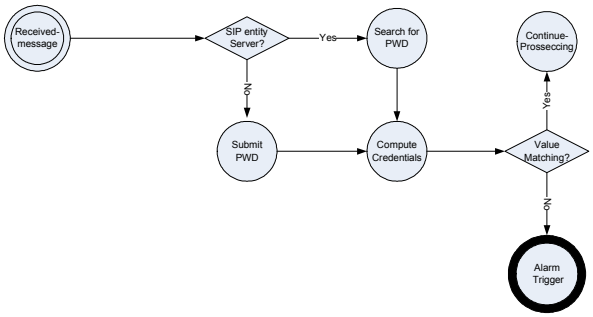

Figure 3. *Integrity-Auth* Scheme Validation Flow Diagram

3.2 An Example of the Integrity-Auth mechanism: The "CANCEL" Signaling Attack

Consider the CANCEL signaling attack described in Section 2, where a malicious user cancels illegally a pending session. Let us assume that the SIP end-user and the corresponding SIP proxy incorporate the *Integrity-Auth* scheme. Under this context, both SIP entities would be able to recognize and discard malicious messages that may affect the stability of the session. Specifically, the malicious user's SIP CANCEL message (see message 4 in Figure 1) will be rejected by the proxy, before any additional processing takes place, as the Integrity-Auth value that will be computed by the SIP proxy will not match the one extracted from the received SIP CANCEL message. This will occur because the malicious user does not have knowledge of the user password which is required to compute the *Integrity-Auth* value.

4. EVALUATION

In order to proceed with an initial evaluation of the *Integrity-Auth* scheme, we have implemented it into:

1. The pjsua SIP User Agent (UA); a SIP soft phone based on pjsip project [11].
2. The SIP Express Router (SER) [12]; a SIP proxy.

As far as the cryptographic operations are concerned, the OpenSSL library [13] has been utilized, while in our configuration the MD5 hash function has been invoked.

The architecture of the experimental test-bed is depicted in Figure 4, while the specific characteristics of the components are given at Table 1.

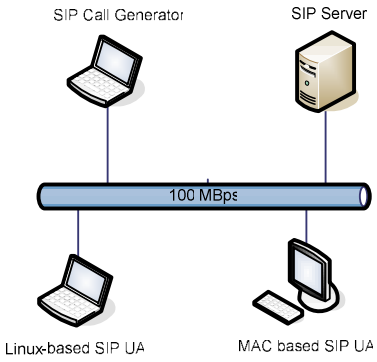

Figure 4. Test-Bed Architecture

Table 1. Test Bed Architecture Specific Characteristic

Network Entity	System Characteristics		
	OS	Processor	Memory
SIP Proxy	Fedora Core 8	Pentium 4 2.8 GHhz	512 MB
Linux Based SIP UA	Fedora Core 8	Pentium 4 2.8 GHhz	512 MB
Mac Based SIP UA	Mac OS X 10.4.11	Intel Core 2 Duo 2 GHz	1GB

Considering that the main objective of this experiment is to identify the overheads, in terms of processing time, introduced by the *Integrity-Auth* scheme, three different scenarios have been implemented. In each scenario different background traffic towards the SIP proxy has been generated (see Table 2 for more details), while the following processing times have been recorded:

1. Metric 1: The overall time until the SIP UA (client) gets an initial response from the SIP proxy
2. Metric 2: The times for all intermediate procedures :
    a. Metric 2.a): Client time for creating and forwarding a SIP request.
    b. Metric 2.b): Client time for validating a SIP response.
    c. Metric 2.c): SIP proxy time for validating a SIP request.
    d. Metric 2.d): SIP proxy time for creating and forwarding a SIP response.

At this point, it should be stated that in order to illustrate potential variations regarding the overhead introduced by different architectures, the SIP UA has been installed in a Macbook laptop as well as in a Linux based personal computer (see Table 1 for system characteristics). Furthermore, for facilitating comparison, all scenarios have been realized with and without embedding the *Integrity-Auth* scheme into the SIP entities.

Table 2. Description of the Scenario

Scenario Number	Description
Scenario 1 (S1)	A legal[1] SIP UA client generates one call per second, without background traffic.
Scenario 2 (S2)	A legal SIP UA client generates one call per second, with additional background traffic towards the SIP Server of 200 calls per second.
Scenario 3 (S3)	A legal SIP UA client generates one call per second, with additional background traffic towards the SIP Server of 400 calls per second.

Figures 5 to 8 depict the overall processing time logged at the client side for scenarios 2 and 3, both for the Linux and the MAC based SIP UAs. Scenario 1 is not shown, since the processing times measured were very close to the times illustrated for scenario 2. Furthermore, Tables 2 to 6 represent the statistical metrics regarding the max, min, average and standard deviation for all scenarios.

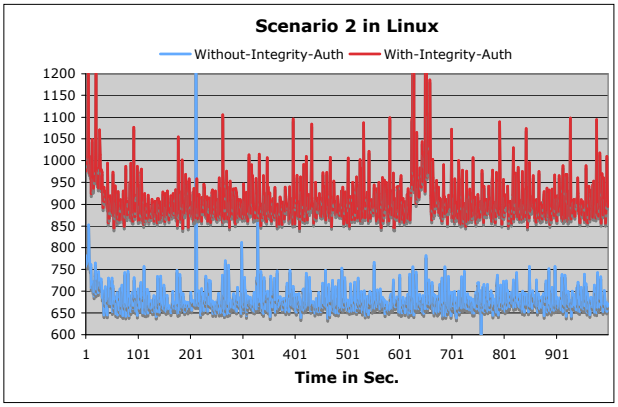

Figure 5. Linux based SIP UA processing time for Scenario 2

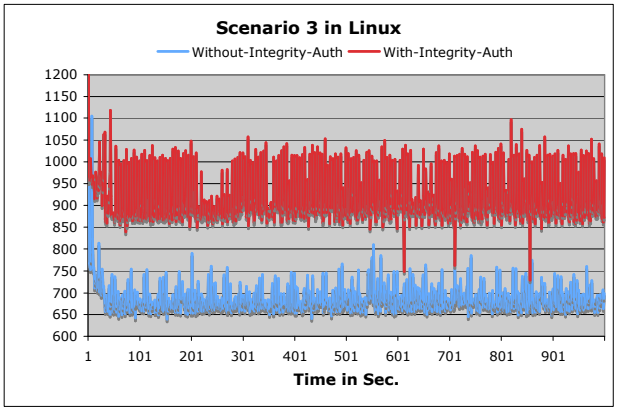

Figure 6. Linux based SIP UA processing time for Scenario 3

[1] Note that in all scenarios the legal SIP UA logs the corresponding processing times (overall time, create request time etc)

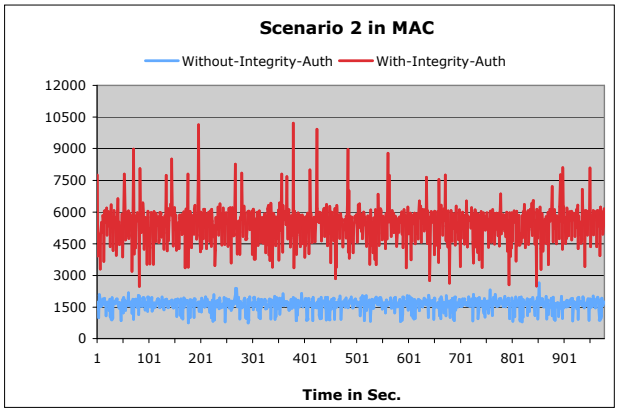

Figure 7. MacBook based UA processing time for Scenario 2

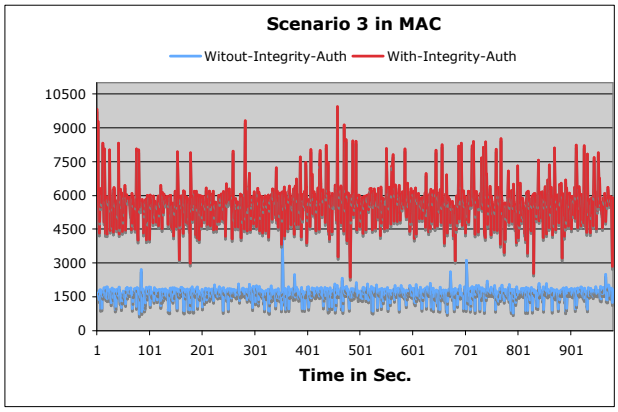

Figure 8. MacBook based UA processing time for Scenario 3

Table 3. Average Statistic Metric for Scenarios 1 to 3

Type of UA	S1	S2	S3
Linux-without-Integrity-Auth	672.70	679.34	684.87
Linux-with-Integrity-Auth	908.51	907.78	910.91
Mac-without-Integrity-Auth	1624.66	1622.88	1635.22
Mac-with-Integrity-Auth	5538.36	5461.17	5641.22

Table 4. Max Statistic Metric for Scenarios 1 to 3

Type of UA	S1	S2	S3
Linux-without-Integrity-Auth	827.00	1375.00	1104.00
Linux-with-Integrity-Auth	1193.00	2223.00	1933.00
Mac-without-Integrity-Auth	2593.00	2641.00	4023.00
Mac-with-Integrity-Auth	10157.00	10192.00	9927.00

Table 5. Min Statistic Metric for Scenarios 1 to 3

Type of UA	S1	S2	S3
Linux-without-Integrity	635.00	557.00	639.00
Linux-with-Integrity	789.00	844.00	729.00
Mac-without-Integrity	751.00	752.00	741.00
Mac-with-Integrity	2014.00	2481.00	2358.00

Table 6. Standard Statistic Metric for Scenarios 1 to 3

Type of UA	S1	S2	S3
Linux-without-Integrity-Auth	19.30	34.72	33.28
Linux-with-Integrity-Auth	49.64	66.98	62.68
Mac-without-Integrity-Auth	252.91	273.98	877.14
Mac-with-Integrity-Auth	831.48	852.15	285.35

Observing the aforementioned processing times it can be deduced that the total average delay introduced by the *Integrity-Auth* scheme for Linux systems is approximately 230 microseconds, whereas for MAC systems is approximately 5000 microseconds (see also Figures 9 and 10). This is mainly due to the different way the two operating systems handle the SIP UA, as no other characteristic of the test bed architecture was modified. Furthermore, it should be stressed that the pjsip-project [11] and the OpenSSL library are designed mainly for Linux based systems, thus affecting the overall system performance when employed in different operating systems.

Regarding the time for creating a request and validating a response, it can be noticed that it varies between the different operating systems that the SIP UA has been installed. In the case of Linux the average time for request generation and response validation is around 120 and 100 microseconds correspondingly, whereas for the MAC the same times are approximately 230 and 290 microseconds; which means almost double the time of the Linux based system. Tables 7 to 10 represent the statistical metrics for the intermediate procedures taking place in these two different SIP UA systems.

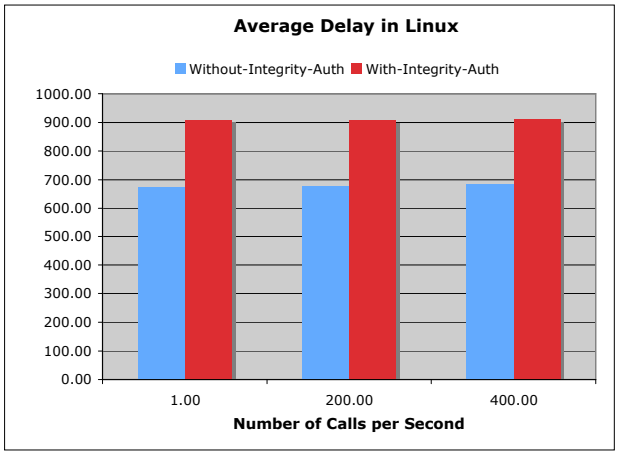

Figure 9. Average Delay introduced in Linux based UA

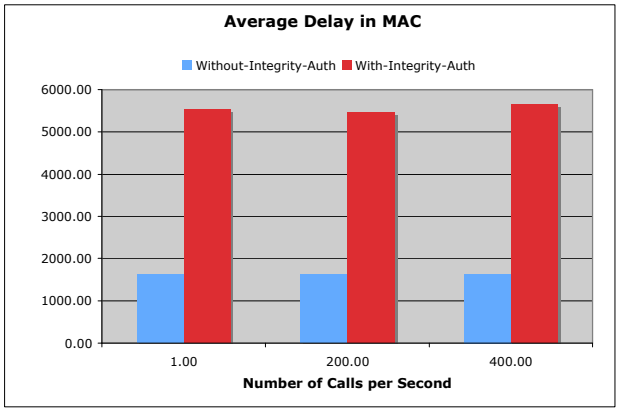

Figure 10. Average Delay introduced in Mac based UA

Table 7. Linux based SIP UA Statistics for Generating a Request

Stat. Parameter	S1	S2	S3
Max	183.00	317.00	407.00
Min	102.00	103.00	101.00
Average	119.33	120.70	117.87
St. Deviation	9.15	10.95	11.65

Table 8. Linux based SIP UA Statistics for Validating a Response

Stat. Parameter	S1	S2	S3
Max	1363	154.00	259.00
Min	95.00	93.00	93.00
Average	109.26	105.55	107.18
St. Deviation	40.88	7.13	18.10

Table 9. MAC based SIP UA Statistics for Generating a request

Stat. Parameter	S1	S2	S3
Max	621.00	494.00	5741.00
Min	174.00	113.00	173.00
Average	225.11	221.62	241.72
St. Deviation	42.02	37.00	182.69

Table 10. MAC based SIP UA Statistics for Response Validation

Stat. Parameter	S1	S2	S3
Max	6632.00	3495.00	3934.00
Min	117.00	116.00	118.00
Average	334.69	274.13	283.92
St. Deviation	594.76	345.11	402.97

Considering the delay introduced by the SIP proxiy in the overall processing time, it is negligible since the average processing time including request validation and the generation of response is approximately 120 microseconds. Tables 11 and 12 show the statistical metrics for the request validation and response generation accordingly.

Table 11. SIP's Proxy Statistics for Request Validation

Stat. Parameter	S1	S2	S3
Max	289	54	452
Min	54	39	37
Average	64.40	46.50	44.15
St. Deviation	18.37	8.10	21.70

Table 12. SIP's Proxy Statistics for Generating a Response

Stat. Parameter	S1	S2	S3
Max	293.00	63.92	237.00
Min	54.00	47.39	40.00
Average	64.41	55.75	45.27
St. Deviation	13.45	8.72	10.75

To sum up, the *Integrity-Auth* scheme proves to be an effective and practical solution for shielding SIP based services against signaling attacks, even for SIP UA that perform "poor" in specific systems, like the SIP UA installed in a MAC system. Assuming the latter's performance as the worst case, the total processing time will not exceed 5000 microseconds. On top of that, it should be stressed that the measured processing times include the time consumed for the logging operation itself. Consequently, the overhead introduced by the integrity-auth mechanism would be even smaller.

5. CONCLUSIONS

ITSs based on SIP should be able to protect all network elements against security flaws and potential attacks. Although, up to now only very few real security incidents regarding ITSs, have been reported, it is almost inevitable that a malicious user will try and eventually succeed to exploit a specific ITS vulnerability. It is therefore crucial to employ the appropriate countermeasures. In this paper we have briefly discussed the problem of signaling attacks as well as an appropriate security mechanism, namely the *Integrity-Auth* mechanism. The *Integrity-Auth* scheme has been experimentally evaluated, in terms of the processing overheads it introduces, in two different operating systems. The derived results demonstrate that the introduced overhead is negligible, providing at the same time adequate protection of SIP based ITSs against signaling attacks.

6. REFERENCES

[1] de Vivo, M., de Vivo, G. O., Koeneke, R., and Isern, G. 1999. Internet vulnerabilities related to TCP/IP and T/TCP. SIGCOMM Comput. Commun. Rev. 29, 1 (Jan. 1999), 81-85. DOI= http://doi.acm.org/10.1145/505754.505760

[2] V. Paxson, M. Auman, S. Dawson, W. Fenner, J. Griner, J. Heavens, K. Labey, J. Semke, B. Volt, Known TCP implementation problems, RFC 2525, March 1999

[3] Geneiatakis, D.; Dagiuklas, T.; Kambourakis, G.; Lambrinoudakis, C.; Gritzalis, S.; Ehlert, K.S.; Sisalem, D. Survey of security vulnerabilities in session initiation protocol, Communications Surveys & Tutorials, IEEE Volume 8, Issue 3, 3rd. Qtr. 2006 Page(s):68 – 81

[4] Butcher, D.; Li, X.; Guo , J., "Security Challenge and Defense in VoIP Infrastructures," IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, vol.37, no.6, pp.1152-1162, Nov. 2007

[5] VOIPSA, VoIP Security and Privacy Threat Taxonomy http://www.voipsa.org/Activities/taxonomy.php, October 2005.

[6] Sisalem, D.; Kuthan, J.; Ehlert, S., "Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms," Network, IEEE , vol.20, no.5pp. 26- 31, Sept.-Oct. 2006

[7] Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Spark, R.; Handley, M;, Schooler, E.; "Session Initiation Protocol", RFC 3261, June 2002.

[8] Yu-Sung Wu, Saurabh Bagchi, Sachin Garg, Navjot Singh, Tim Tsai, "SCIDIVE: A Stateful and Cross Protocol Intrusion Detection Architecture for Voice-over-IP Environments" In Proceedings of the 2004 International Conference on Dependable Systems and Networks

[9] Feng Cao, Cullen Jennings, "Providing Response Identity and Authentication in IP Telephony", in the proceedings of First International Conference on Availability, Reliability and Security, April 2006

[10] Dimitris Geneiatakis, Costas Lambrinoudakis, "A lightweight protection mechanism against signaling attacks in a SIP-based VoIP environment", Telecommunications Systems, Springer, Vol 36 no.4 pp153-159, February 2008

[11] PJSIP - Open Source SIP Stack available on http://www.pjsip.org/

[12] SIP Express Router (SER), available on http://www.iptel.org

[13] OpenSSL Library, available on http://www.openssl.org