Int. J. Inf. Secur. (2013) 12:201–217
DOI 10.1007/s10207-012-0187-0

REGULAR CONTRIBUTION

An intrusion detection and prevention system for IMS and VoIP services

Nikos Vrakas · Costas Lambrinoudakis

Published online: 10 January 2013

Abstract The Voice Over IP (VoIP) environments and the most contemporary ones such as the IP Multimedia Subsystem (IMS) are deployed in order to provide cheap and at the same time high quality services to their users. Video calls, conferences, and applications can be provided to mobile devices with the lowest possible delay, while the Quality of Service (QoS) remains as the top priority for users and providers. Toward this objective, these infrastructures utilize the Session Initiation Protocol (SIP) for signaling handshakes since it is the most flexible and lightweight protocol available. However, according to many researches, it happens to be vulnerable to many attacks that threaten system's security and availability. In this paper, we introduce a cross-layer mechanism that is able to mitigate in real-time spoofing attacks such as SIP signaling, identity theft, masquerading, and Man in the middle, and also single and distributed source flooding. It consists of three components: the policy enforcer which acts as a black list, and the spoofing and flooding modules. We also introduce a classification of SIP flooding attacks for better representation of the detection coverage. To the best of our knowledge, the proposed detection system is the most complete and accurate in terms of the attack range that is able to deter. Concerning its performance, it does not require computational expensive calculations nor resource demanding security protocols, thus being a lightweight mechanism. The experimental results have demonstrated high detection rates with false alarm rates approaching zero. Finally, it is platform independent and transparent to networks' operations and thus can be deployed in both VoIP and IMS environments.

Keywords Attacks · SIP · IMS · Security · IDS · Intrusion · Flooding · Spoofing

1 Introduction

The multimedia services that are offered through the internet have become an inseparable part of people's life. In addition, technological developments have enabled the provision of such services through mobile and handheld devices. This is achieved through the IMS deployment [1]. The high resource demanding services that IMS provides, such as video conferences, audio calls, applications, IP television, and many more, must be streamed with high Quality of Service (QoS). Considering QoS, these infrastructures are employing a lightweight signaling protocol; SIP [2]. This text based protocol is flexible enough to easily incorporate and provide different services. It is also a low resource demanding protocol without burdening the infrastructure with further delays during the session establishment handshakes.

However, together with the advantages, there is also a drawback; there are many security vulnerabilities that can be exploited by malicious internal or external users in order to degrade the QoS causing Denial of Service (DoS), intercept the communication sessions and steal user's identities and credentials. Moreover, the attacker can utilize techniques from the lower layers of the internet protocol stack in order to threaten SIP services. For instance, IP spoofing [3] or Address Resolution Protocol (ARP) poising [4] can be the first step of an attacker aiming to manipulate a SIP request. Every architecture that utilizes SIP as its signaling protocol

N. Vrakas · C. Lambrinoudakis (✉)
Department of Digital Systems, University of Piraeus,
150 Androutsou St., Piraeus18532, Greece
e-mail: clam@unipi.gr

N. Vrakas
e-mail: nvra@unipi.gr

is susceptible to such behaviors. Many scientific works pinpoint these vulnerabilities [5–7].

In VoIP and IMS environments, different security protocols are deployed for hardening the defense against the above-mentioned behaviors. For instance, IMS can utilize Authentication and Key Agreement (AKA) with IPSec [8], or the SIP Digest with TLS [8]. These protocols provide authentication, confidentiality, and integrity services to the communication. Also the SIP Digest can be utilized for low resource enabled devices [9] but it provides only authentication support to SIP messages. Nevertheless, these mechanisms can prevent most of the attacks originated by external users but they cannot effectively discourage malicious subscribers to launch flooding or SIP signaling attacks through their security tunnels.

Many researchers work toward the detection of such security incidents but most of the published results cover only a small subset of the attacks [10–12] or are limited to the detection of the attacks without being able to prevent them [12–14] or utilize [15] heavy weight protocols such as Public Key Infrastructures (PKIs) that introduced significant delays.

Other solutions such as Transport Layer Security (TLS) and Secure Multi-Purpose Internet Mail Extensions (S/MIME) cannot deter internal flooding and signaling attacks and on top of that they introduce significant overheads [16,17].

This paper presents, to the best of our knowledge, the most comprehensive and thorough cross-layer mechanism for the detection and prevention of a wide range of attacks against SIP-based environments. Furthermore, it is considered lightweight since it does not employ processing consuming security protocols nor expensive mathematical calculations. Finally, it is transparent to system's and user's operation, it does not impose any modifications to them, and it can be easily deployed in both IMS and VoIP infrastructures. Specifically, all the messages are first checked for their authenticity using data gathered from layers 2, 3, and 5 of the internet protocol stack. If the SIP registration message is authenticated, then a bind is created which correlates information of the aforementioned encapsulated protocols. This bind is stored into a table with the help of a bloom filter. All the incoming messages are checked against the entries of this table thus detecting all spoofed messages. However, non-spoofed messages are not always legitimate since there are flooding attacks which can be launched without forged SIP request. Thus, a second statistical module is utilized that detects deviations among all bindings with respect to their traffic behavior. When a binding satisfies a rule that has been created during the training period of the mechanism, then the corresponding messages are dropped and their sources are blacklisted. It is worth noting that the mechanism is not restricted in detecting only INVITE or REGISTER message floods but it can also effectively deter constant and increasing rate floods launched with any of the SIP's available request methods. A classification of flooding attacks in VoIP/IMS environments is also presented in order to highlight all the different cases that the intrusion detection and prevention systems have to confront.

The paper is structured as follows: Sect. 2 briefly describes flooding and spoof-based attacks in VoIP/IMS environments. Sect. 3 provides a classification of the flooding attacks, while Sect. 4 presents the proposed mechanism focusing on the justification of its detection accuracy. Section 5 includes a review of other researchers' proposals in the field, as well as a comparison of the different detection mechanisms. The paper concludes with some assumptions and thoughts for future work.

2 Attacks against SIP/IMS infrastructures

The SIP protocol is utilized for session establishment and handling in IMS and in the majority of VoIP infrastructures. The loose syntactic rules and the text-based format of the messages comprise a lightweight and flexible protocol that succeeds high Quality of Service (QoS) with low response times. Nevertheless, these features also render the protocol vulnerable to various attacks and security breaches [18–21]. The employment of a security mechanism such as HTTP Digest [9] may deter the attacks originated from external attackers but not from internal malicious users. The same applies to IMS infrastructures. The employment of an authentication mechanism such as AKA with IPSec [8] can neither prevent threats originating from Internal Attackers (IAs) since they can launch attacks through their legitimately established IPSec tunnels.

Moreover, a successful attack may involve the compromise of different layers of the internet protocol stack, such as the network or the data link layer. For instance, an attacker may launch an ARP poisoning [22] attack in order to gather the Authentication Vectors (AV) from a handshake, breaking the authentication mechanism [21] or intercepting the communication [23].

2.1 Forged message attacks

Threats in VoIP/IMS environments may involve the manipulation of layer 2, 3, or 5 messages. For the application layer, the main attack categories are: SIP signaling manipulation, masquerade, Man in the Middle (MitM), and replay attacks.

In signaling attacks, the attacker utilizes SIP protocol's requests in order to cause DoS to the server or to a specific user. The CANCEL and BYE requests are responsible for revoking or terminating multimedia sessions, respectively. Spoofing the headers "From" and "Call-id" of such requests, an attacker can terminate a session illegally. This attack can

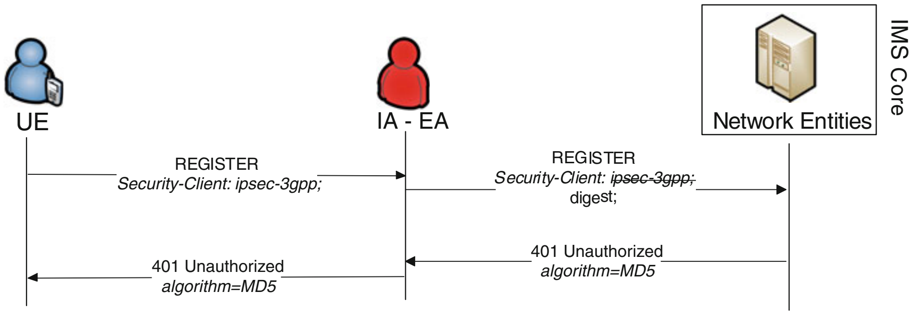

Fig. 1 Security level degrading attack. The attacker impersonates both the UE and the home network by launching DNS or ARP poisoning techniques. The stronger security mechanisms are removed during the negotiation with the proxy

be launched through the security tunnels by an IA especially in case of a weak parser's implementation. Another DoS attack can be launched utilizing UPDATE or re-INVITE requests. Specifically, the malicious user is able to mute a multimedia session or launch a hijacking attack as described in [10]. Authentication and integrity mechanisms may deter the External Attackers (EAs) but they do not always provide a comprehensive solution against malicious subscribers.

In masquerade attempts, an attacker's objective is to impersonate a specific User Equipment (UE) or even a user. These attacks are known as SIP spoofing or identity theft correspondingly. The attacker includes a stolen IP Multimedia Public Identity (IMPU) (or private identity–IMPI) to his messages instead of his real one, in order to charge the provided services to victims' identities. Thus, the IA is charged only for the IP connectivity (during the IP allocation from the GGSN) and not for the multimedia services provided by the IMS. The employment of authentication and integrity mechanisms in messages or security tunnels cannot guarantee the discouragement of such behaviors. A masquerade attack can also be applied in the third layer where the attacker spoofs the 32-bit string of the source IP address header of the packet in order to bypass the SA-SIP check: A correlation between the IP and the given public ID of the messages derived from the SAs which have been established during the AKA in the registration handshake. This procedure is executed by the S-CSCF (in VoIP architectures such checking procedure is not implemented since is not described in the specifications).

In a MitM attack, malicious users are placed in the middle of the communication path between the user and the server [24–26]. In this type of attack, the attacker bypasses both integrity and authenticity security requirements and consequently is able not only to impersonate users or network elements, but also to gain unauthorized access to the provided services, intercept the communication channel or even to cause denial of service. These attacks can be launched by utilizing either ARP poisoning [4] (in layer 2) or Domain Name System (DNS) poisoning (in layer 5) [22] techniques. The attacker changes the IP-MAC or the domain-IP associations correspondingly in order to redirect the traffic through him (acting as gateway) and gathers communication channel's data.

In fact, in VoIP/IMS infrastructures, after an ARP or DNS poising attack follows a SIP-based attack where the messages are manipulated, imposing further damage to the system. For instance, the attacker may spoof the '*expires*' header of a registration request to zero causing an immediate deregistration of the victim [27]. Another attack can be launched after a successful MitM by downgrading the security level of the upcoming session (Fig. 1).

Specifically, during the session establishment handshake, the intermediate manipulates the header (security-client value in authorization) that includes the available security suites and removes the stronger ones [28]. Thus the S-CSCF (or the SIP server) will inevitably choose one of the weak security protocols that the attacker has left available in the header. Usually, a different attack will follow since the attacker will be able to break the employed security mechanism. Also a conference interception could be the result of a MitM attack between the user and the MRFC/AS. An IA spoofs the header Refer-to or Refer-by of the gathered messages, in order to silently invite himself in a conference room [23]. Finally, a MitM can lead to abuse of the authentication mechanism. As described in [21], the attacker acts as intermediate between the proxy (or the P-CSCF in IMS environments) and the user, masquerading both of them, managing to steal the AV in order to authenticate his messages. This attack concerns only the SIP Digest authentication mechanism.

In replay attacks, a malicious user initially simply observes and captures the signaling data between a legitimate user and its home network. He focuses on capturing authenticated messages in order to craft spoofed requests/responses that include the authentication vector of a legitimate user and thus facilitating the attacker to impersonate the legitimate user and get access to his services. More details on spoofing-based attacks can be found at [29].

2.2 Flooding attacks

Flooding attacks in SIP-based environments such as VoIP and IMS are similar to attacks taking place in the transport layer and the Transmission Control Protocol (TCP) [30]. More precisely, when a client needs to establish a TCP connection with another host, a three-way handshake is required.

The client sends a synchronization number (SYN) to the server; the latter allocates memory resources for a specific amount of time and responds with a SYN-ACK containing the received SYN incremented by 1. Finally, the client acknowledges the reception with an ACK including the SYN-ACK incremented by 1. At this point, connection is established and the server releases the allocated resources. If the client spoofs its source IP address with a non-existing one, the server will never get the final ACK and thus the allocated resources will not be released. If the client forward a large amount of such requests, the server will soon run out of memory resources causing DoS [31].

This can be applied during a session establishment in SIP. For instance, the attacker may spoof the contact header of an INVITE request. The server (or P-CSCF in IMS) will allocate memory resources for session handling. While the contact header points to another IP, the server will not receive the SIP ACK request in order to release the memory resources. A flood of such requests may lead to DoS as described above. This is called SYN syndrome attack since it is based on the TCP's SYN vulnerability. The attacker can also launch a combined forged message and flooding attack: He can forge different messages with randomly chosen IP addresses (Fig. 2) enhancing the impact of the SYN syndrome attack (new memory allocation for every SIP uri/IP).

Another case of flooding attack can be launched by forwarding an enormous amount of requests to a specific network entity in order to cause large delays in active sessions or in the session establishment procedures. The target can be the P-CSCF or the MRFC/AS or even a UE. The latter can be easily flooded due to their limited CPU and memory resources and the maximum incoming traffic that they can handle. The Distributed Denial of Service (DDoS) attacks are launched against the more hardcore network entities. The attacks are originated by multiple sources which forward SIP requests with high rates, draining the memory and CPU resources of the target system. The attackers can be innocent UEs/servers or attacking networks. In the case of innocent UEs or servers the attacker may have infected them with malicious software turning them into zombies (called slaves or zombies because they execute orders as dictated by the master namely the attacker). Then, the attacker can deploy all of them at the same time in order to flood a target machine introducing large delays in sessions.

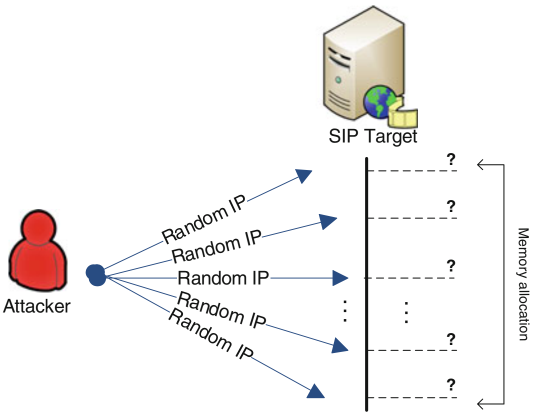

Fig. 2 The SIP-based network entity is flooded with spoofed messages. The attack target allocates resources for each message until the reception of a response

Another case of flooding which involves innocent entities is the INVITE reflection syndrome [13]. The attacker sends many INVITE requests to different servers/UEs with spoofed SIP contact header. Thus, all the involved servers will respond to the given IP of the contact header, namely the target machine causing DoS circumstances.

A replay attack can also be used for flooding network entities in environments where the message authentication is mandatory. The attacker forges the messages with (random or fixed) IP addresses and includes the victim's valid authentication string, bypassing the authentication. The responses do not depend on IP addresses since they are calculated through the following function: $H(H(\mathrm{A1}), \text{nonce ":" } H(\mathrm{A2}))$, where H is a hash function, A1 = username ":" realm ":" password and A2 = Method ":" digest-uri-value.

Finally, a CPU flooding attack can be launched utilizing registration requests (Fig. 3). The malicious user can force the IMS core or the registrar server in VoIP infrastructures, to execute cryptographic functions which are considered computationally expensive, in order to validate all the incoming requests (i.e., for every new registration message, the IMS core must accomplish at least: the detection of an appropriate S-CSCF and the computation of new/fresh AVs). The specific attacks, in addition to the fast memory consumption due to numerous half-open connections, are mainly focused on consuming target system's processing resources. Especially in IMS environments, they can introduce delays to network due to the heavy weight employed security mechanisms and the large number of network entities that are involved in every registration request (i.e., opens many diameter connections over Cx, Dx with HSS and Gq interfaces with PDF).

Further description can be found in the next section (Sect. 3), while a more detailed review of flooding attacks in SIP can be found in [13].

3 Flooding attacks' classification

This section provides a classification of flooding attacks and thus facilitates the analysis of the attacks in conjunction with the proposed Intrusion Detection and Prevention System (IDPS). The behaviors have been classified in terms of (1) the type of the messages (registration or other type

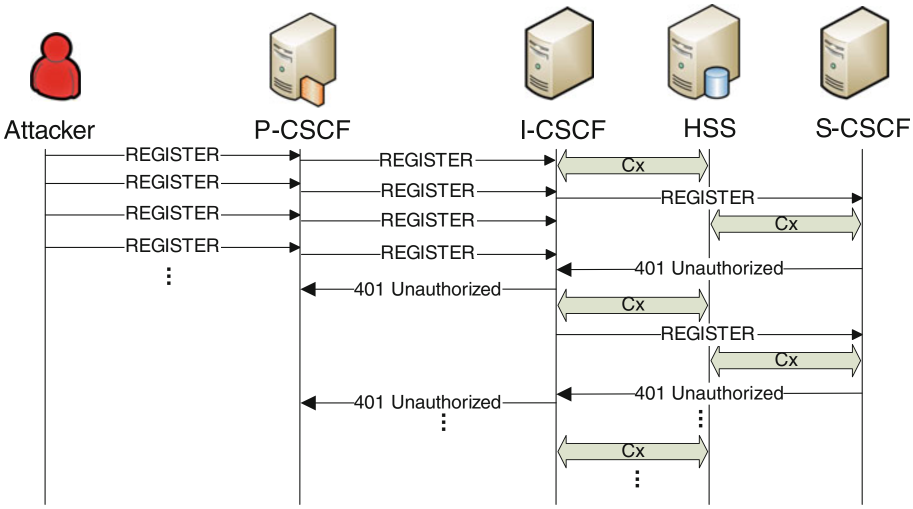

Fig. 3 Flooding attack in IMS. Every request involves many network entities and calculations

of requests) and (2) the access level of the attacker. Additional parameters that have been considered are whether the attacks (3) are launched from single or multiple sources, (4) use messages that are spoofed with fixed or different IP addresses, and (5) include or not authentication vectors. Through this analysis, it is possible to deduce the attacker's objective and also the range of the attacks that be mitigated.

Concerning the access level, an attacker can be Internal (IA) or External (EA). An IA is the entity that has a legitimate subscription to the IMS services and uses its credentials in order to launch flooding attacks. Especially in IMS environments, they can launch attacks through the security tunnels when AKA with IPSec [8] is employed. On the other hand, the EA does not have any legitimate subscription to the server. However, the latter can launch flooding attack since the first registration request is sent without any protection both in VoIP and IMS environments according to specifications [2] and [1], respectively. Furthermore, we discriminate between registration messages and the rest of SIP requests due to their crucial difference: The registration requests are sent without security protection, while almost all others require authentication such as IMS AKA with IPSec [8], SIP Digest [9] or SIP Digest with TLS [8]. Such attacks can be launched with spoofed or legitimate messages.

Identifying whether an attacker uses a fixed SIP-IP or not (the same attacker utilizes the same SIP message with variable "from" header) enables the determination of the type of the attack and its target (e.g. resources exhaustion, brute force, end-user flooding). These different categories can be further divided relating to whether a message contains a valid authentication string or not and consequently to conclude the target of the attack (e.g., CPU resources exhaustion, end-user flooding, etc.). Finally, there are two main types of DoS attacks: the distributed (DS) and the single source (SS). This final classification is of major importance because the attacker can "silently" flood the servers utilizing lower rate flooding attacks from different machines (zombies) at the same time. Figure 4 depicts the above-mentioned classification tree of flooding attacks. The circles denote the target and the specific type of the attack.

3.1 Internal attacker

A legitimate subscriber may act maliciously (as an IA) by launching flooding attacks from SS. The utilized messages can be either registration or non-registration requests. If the attacker uses different spoofed IPs in the malicious messages, then he probably launches a SYN syndrome attack or a resources exhaustion attack (see Sect. 2) by forcing the server to open an enormous amount of encrypted connections (over Dx, Cx) with the HSS and challenge string calculations. This observation is based on the fact that the attacker avoids the reception of all responses and also tries to allocate server's memory resources per different IP address.

In the case where the attacker utilizes spoofed registration messages with the same IP address in all messages, it can be deduced that he launches a brute force attack trying to break user's password or resources exhaustion attack as formerly noted, utilizing a powerful attacking server. This is based on the fact that the attacker needs to gather the server's 401 and 200 OK responses.

Another SS attack can be launched by an attacker utilizing other SIP signaling messages such as the INVITE request. Depending on the type of the DoS, the attacker decides to spoof or not the specific INVITE message. When an end-user is the target of the attack, the malicious signaling traffic must be authenticated so the attacker does not spoof the IP addresses. Therefore, an enormous amount of apparent legitimate (the originator is a subscriber and the messages have been authenticated) traffic reaches the

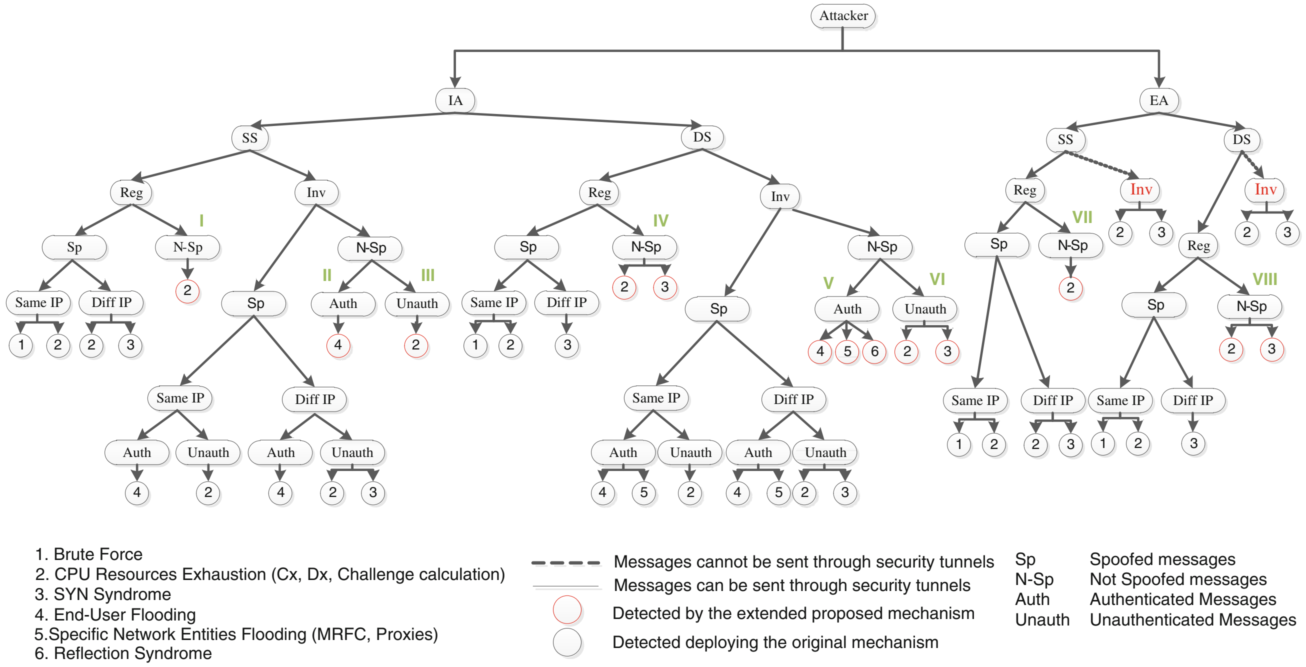
Attacker
IA
EA
SS
DS
Reg
Inv
Sp
N-Sp
Same IP
Diff IP
Auth
Unauth
I
II
III
IV
V
VI
VII
VIII
1. Brute Force
2. CPU Resources Exhaustion (Cx, Dx, Challenge calculation)
3. SYN Syndrome
4. End-User Flooding
5.Specific Network Entities Flooding (MRFC, Proxies)
6. Reflection Syndrome
Messages cannot be sent through security tunnels
Messages can be sent through security tunnels
Detected by the extended proposed mechanism
Detected deploying the original mechanism
Sp Spoofed messages
N-Sp Not Spoofed messages
Auth Authenticated Messages
Unauth Unauthenticated Messages

Fig. 4 Flooding attack's classification tree

user's UE causing DoS. If the messages are not authenticated, they can also lead to DoS as a resources exhaustion attack.

All the above-mentioned cases of attacking possibilities exist and can be launched from multiple sources either as an invite reflection or zombie attacks (see Sect. 2). Actually, in terms of classification, SS and DS differ only in the effects they can induce to the network/servers due to the massive amount of traffic that the latter can utilize. Therefore, when the attacking entities forge registration messages, they can achieve not only a resources exhaustion attack but also a SYN syndrome attack. The reason is the enormous amount of different requests (i.e. different IP address per request) that reach the servers.

Also in DS attacks, where invite or other non-registration requests are utilized, the core network entities are also threatened due to the large volume of traffic and resources deployed by the attackers. If the requests are deliberately unauthorized, they can cause resource exhaustion to the core servers, as in SS attack, and also a SYN syndrome effect (many IP addresses). The same is true for not spoofed SIP messages with random IP addresses. Moreover, a reflection syndrome can be achieved with not spoofed messages because the innocent servers/UEs are involved without any spoofed headers; note that only the originator (namely the attacker) spoofs the contact header. This is also the case when the messages are spoofed with fixed IPs with the exception of unauthenticated ones: the SYN syndrome in this case is not possible because numerous messages reach the servers but with the same IP; hence, no more memory resources will be allocated.

3.2 External attacker

An EA can launch flooding attacks both from SS or DS. Basically, an EA does not have a subscription to the server, and thus, many of the attacks cannot be launched. As a matter of fact, when AKA IPSec is employed, the tunneling is mandatory for every session between the P-CSCF and UE except for the first registration message, as the IMS's specifications [8] defines. Therefore, non-registration requests can be forwarded to the proxies only when SIP Digest, GPRS-IMS-Bundled Authentication (GIBA) [32], or the NASS-IMS-Bundled authentication (NIBA) [33] are employed. These cases are represented with dotted arrows in Fig. 4. Moreover, the non-registration requests originated by an EA can only be spoofed and unauthenticated. The authenticated ones fall into the IA's category. Therefore, either SS or DS flooding attacks can lead to DoS by exhausting CPU's resources or by a SYN syndrome if the IPs have been chosen randomly.

As described in the previous section (see Sect. 2), an EA can flood a target with stolen authentication strings (replay attack) when SIP Digest is employed, since the calculation of responses on this security protocol does not depend on IP addresses. If the replay attack is not launched or it is not successful, the large volume of unauthenticated traffic can lead to CPU resources exhaustion in the case of fixed IP addresses. Otherwise, the randomly chosen IP addresses add the probability of a SYN syndrome attack through a large amount of half-open connections (server maintains a specific amount of memory per IP for a predefined and fixed period of time—see Fig. 2).

However, registration messages can be forwarded from an EA and processed by the signaling core due to the lack of authentication requirements. Taking into account the aforementioned facts about the registration messages, it can be deduced that their effects on the architecture are exactly the same in both cases, irrespective of whether the attacker is internal or external.

4 Proposed mechanism

According to Sect. 2, threats in SIP/IMS environments may originate from different layers, since the attacker attempts to exploit more than one of the protocol's vulnerabilities. It is therefore important for the IDPS to be able to detect such behaviors before they become a threat for the higher layers and affect multimedia services. The proposed IDPS is based on the concept of the cross-correlation table that has been presented in [23,29]. More specifically, a table is being employed in order to maintain specific information for every registration request from layers 2, 3, and 5 of the protocol stack with every row maintaining the specific requests' values after their successful registration. Table 1 presents a simplified instance of such a table. The columns "*C*" and "Variables/threshold" are employed only by the extended mechanism (ext.) and are utilized for detecting flooding attacks. A detailed description can be found later in this section. A decision tree has been also introduced in order to facilitate the discrimination between fake and legitimate messages.

The proposed mechanism consists of two distinct modules: the first one is responsible for registration requests while the other module for all remaining requests. It has been designed not only to prevent spoofing attacks, such as SIP signaling, UE, and User ID impersonation, MitM, but also attacks against system's availability known as flooding attacks. It is true that many flooding attacks utilize spoofed messages and they can therefore be mitigated. However, they can be also launched without spoofed messages, for instance though a security tunnel. Considering Fig. 4, we can pinpoint (red circles) eight cases of flooding attacks that the mechanism presented in [29] is not able to deter:

(I) If a UE and a user id have never been registered before, namely it is the first registration message, this message will be dropped. This happens because there is not any record for this specific combination. Thus, the UE cannot be registered. On the other hand, if the IDPS gives unprotected access only for registration messages, the architecture will be vulnerable to flooding attacks.

(II) In case of an IA who authenticates all his messages without forging them, neither a conventional firewall nor a security mechanism such as IPSec can detect them during a flooding attack.

(III) These requests are not spoofed but they are unauthenticated. Taking into account that the IA is already registered, he can utilize his legitimately established security tunnel to send them. According to specifications [8], after the tunnel establishment, authentication vectors are not included in SIP messages. Hence, flooding attacks in such a case cannot be deterred.

(IV) This case is similar to I

(V) This case is similar to II

(VI) This case is similar to II

(VII) and (VIII) These cases are similar to I

The above-mentioned attacks are addressed by the proposed extended IDPS. More specifically, all cases depicted in Fig. 4 can be detected and mitigated through the deployment of the second module (Module II).

4.1 Mechanism description

The proposed mechanism consists of two modules (Fig. 5). The first module handles every incoming message, takes decisions about its originality, and decides whether it will be routed it to its destination or not. The main concept is based on cross-layer binding between six values that can be gathered from layers 2, 3, and 5 of the network protocol stack. These values correlate a specific UE with a session, a set of IP

Table 1 A simplified instance of the proposed mechanism's cross-layer correlation table

	C	UE	IP address		IMPI/IMPU	Method	Variables/threshold
E_0	Counter	MAC_{12}	IP_{12}	SIP_{12}	ID_{12}	REGISTER	Flooding detection: variables/thresholds
E_1		MAC_{24}	IP_{24}	SIP_{24}	ID_{24}	REFER	
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
E_n		MAC_n	IP_n	SIP_n	ID_n	INVITE	
		Layer 2	Layer 3	Layer 5			
	Ext.						Ext.

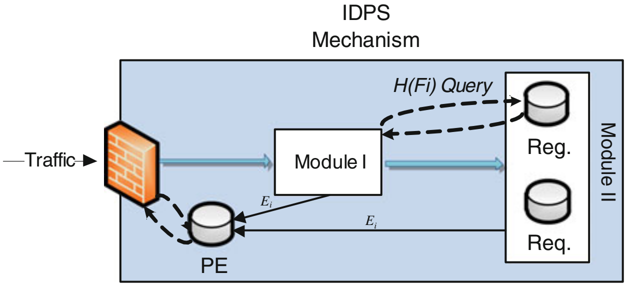

Fig. 5 Proposed mechanism's architecture

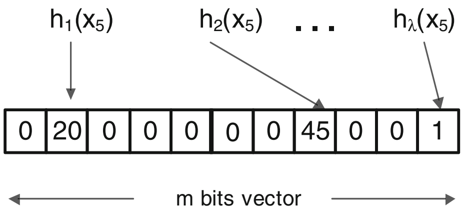

Fig. 6 Counting bloom filter

addresses, and the identities of the subscribers. For instance, the frames located at the data link layer (layer 2) bear the network or MAC address of the utilized UE. Furthermore, the binding among the IP address of the 3rd and 5th layer and the MAC address must be unique at a specific point of time.

For every incoming message, a tuple E_i, $\forall i \in \{0, \ldots, n\}$ is generated, where n is the number of incoming messages and $n \in N$.

Every E_i passes through the spoof checking module which decides whether the message is legitimate, and thus, whether it will be forwarded to the second module or it will be dropped according to some rule of the Policy Enforcer (PE). The latter holds a blacklist of known malicious E_i. An incoming message is firstly checked for existence in the PE's list. Only the non-listed messages should be forwarded and handled by the next modules.

Afterward the legitimate E_i is fed to the second module which consists of two tables: the registration table for holding registration messages' data and the request table that holds the data of all the other requests. Every E_i is stored to one of these tables, depending on the type of the request.

The position of the table where a tuple must be stored is calculated according to the the bloom filters theory [34]. A bloom filter is a data structure that can be utilized for testing the existence of an element x_i in a set X. Every x_i is hashed through λ different hash functions. The result of every hash function points to a specific position of a vector of m bits. The vector's length is equal in bits as the output of the hash functions. Initially, all the vector's bits are zero. When a hash output points to a specific position, it turns the null bit to 1. A x_i exists in the set/vector when all λ outputs point to positions with all of them having a value of 1.

A more advanced bloom structure is the counting filter. The conventional filter does not allow any other operations (e.g., subtraction or summation) than simply turning the zero bits to 1. The counting bloom filters are capable of counting how many times a hash output points on specific table position (Fig. 6).

This model is being employed in order avoid searching and sorting through the second module's tables: The first column in both tables is a counting bloom filter. The input to the hash functions is the tuple $F_i = \{\text{MAC}_i, \text{IMPI}_i, \text{IP}_i\}$, $F_I \subseteq E_i$. These three values denote a unique combination that is extracted from every E_i.

Therefore, the precise position for every user per UE/IP on the tables is the value H(F_i). Both of the tables have eleven different columns as depicted in Table 2: C is the counting bloom filter that provides the mechanism with information about the number of messages per subscriber/UE and eliminates the time needed for detecting a specific tuple's position. The values MAC, IP, SIP-IP denote the corresponding address at layer 3, 4, and 5 of the network protocol stack that have been involved in a specific request. The IMPI/IMPU holds the private/public id of the corresponding incoming message. The TS holds a timestamp and T.Dist the time distance between the last two timestamps that has been calculated by: $T.\text{Dist} = \text{TS}_i - \text{TS}_{i-1}$, where i is the number of messages that were stored in a specific row. Init.D_Avg and Curr.D_Avg denote the initial and current T.Dist value, respectively. Finally, the Trs value is a threshold for alarm triggering in single source flooding attacks. The purpose and use of these values are described in Sect. 4.2 below. The monitoring method is depicted in Fig. 7, while Fig. 8 provides the pseudo-code of the monitoring procedure and of

Table 2 Requests' table

		C	MA	IP	SIP-IP	IMPI/IMPU	Method	TS	T. Dist	Init. D_Avg	Curr. D_Avg	Trs.
$H(F_{74})$	→	54	MAC_{11}	IP_{11}	$SIP\text{-}IP_{11}$	clam	INVITE	100	15	14	10	1
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$H(F_{42})$	→	20	MAC_4	IP_4	$SIP\text{-}IP_4$	nvra	NOTIFY	10	5	7	5	5
		60	MAC_7	IP_7	$SIP\text{-}IP_7$	unipi	INFO	1,500	12	8	4	7

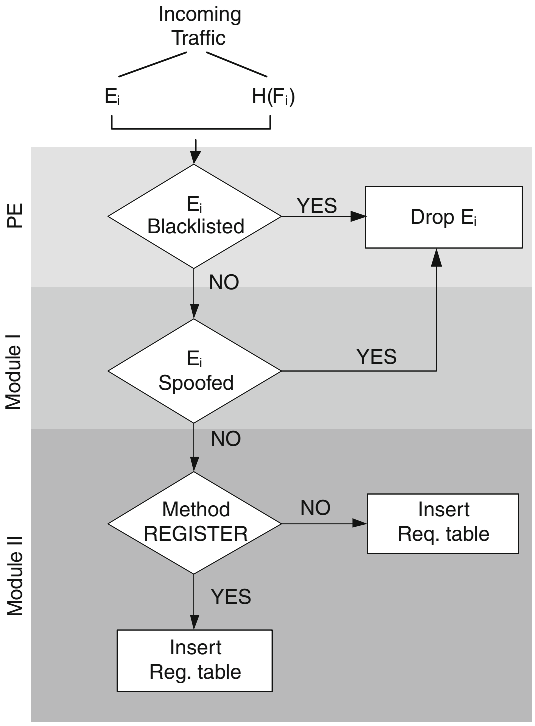

Fig. 7 Monitoring method

the way the incoming messages are handled by the proposed mechanism's components.

4.2 Spoofing detection method

In order to provide a more accurate and descriptive presentation of the proposed mechanism's spoofing detection procedure, we define the following sets for every one of the network layers involved: $M = \{\text{the set of MAC addresses}\}$, $I = \{\text{the set of IPs}\}$, $S = \{\text{the set of SIP-IPs}\}$, $D = \{\text{the set of IMPIs and IMPUs}\}$, $A = \{\text{the set of SIP methods}\}$. Moreover, if $R = \{\text{REGISTER}\}$ then $R \subseteq A$. Therefore, $E_i = \{m_i, i_i, s_i, a_i, d_i\}$, where $m_i \in M$, $i_i \in I$, $s_i \in S$, $a_i \in A$ and $d_i \in D$. Also, $K_i = \{m_i, i_i, s_i, d_i\}$ and $O = \bigcup_{i=1}^{n} K_i$ and finally $W = \bigcup_{i=1}^{n} E_i$.

For every incoming message, the values E_i and $H(F_i)$ are generated. The spoofing module (module I) detects the position $H(F_i)$ on the registrations' table and retrieves the corresponding C_i's data. Let E_c be the corresponding tuple. The mechanism proceeds with the execution of the following procedure between tuples E_i and E_c (Fig. 9).

If $E_i \cap E_c = \emptyset$ and $a_i \in R$, then E_i corresponds to a new registration procedure; therefore, there is no identical set in W. If the specific message has been authenticated and $i_i = s_i$ (intra-packet check), then the corresponding tuple will be forwarded to module II. There, it will be stored in the registration table, denoting the first registration and the binding of the specific UE-subscriber for this specific period of time.

If $E_i \cap E_c = \emptyset$ and $a_i \notin R$, then there is no identical set in W; thus, the message has been spoofed and shall be dropped. The PE has to be updated as well with the E_i tuple. The UE is actually not yet registered and this is derived from the fact that the two (E_i, E_c) sets do not have common elements.

If $E_i \cap E_c \neq \emptyset$, then at least one of the $m_i, i_i, s_i, a_i, d_i \in E_c$.

(i) Let only $m_i \in E_c$, then $i_i, s_i, a_i, d_i \notin E_c$. Therefore the corresponding message shall not be processed because this corresponds to an identity theft attempt

Fig. 8 Mechanism's message handling pseudo-code

```
INPUT: The extracted Ei tuple from incoming traffic
OUTPUT: An Ei that shall be blacklisted or a
legitimate Ei that shall be inserted to the second
module's table

modulei(Ei);

    if module1 == 1 /* module1 returns 1 when
    message shall be routed and 0 otherwise*/
    then
        Hash = H(Ei.MAC||Ei.IP||Ei.ID);
        if Ei.method = REGISTER
            insert_reg_table(Hash, Ei);
        else
            insert_request_table(Hash, Ei);
    else
        drop_packet(Ei);
        upadate_PE(Ei);
```

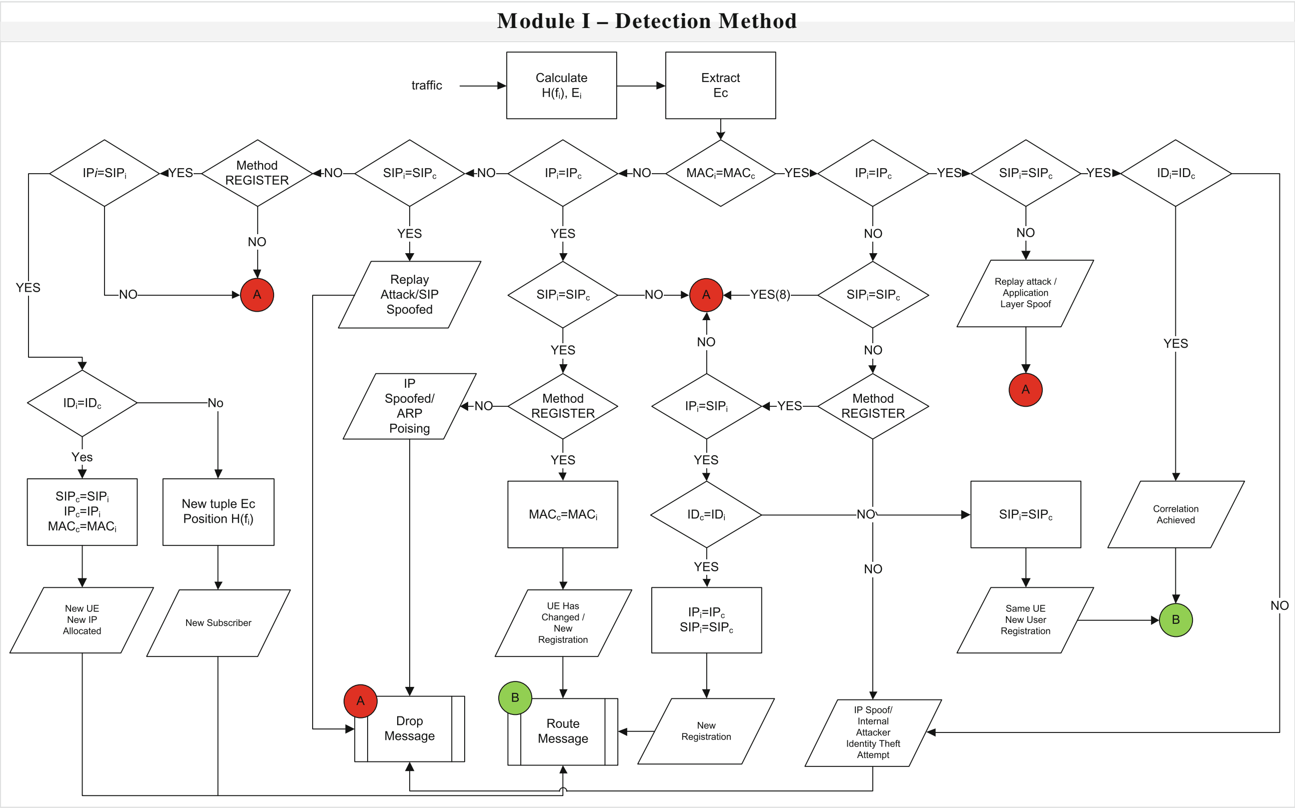

Fig. 9 Module's I spoofing detection method

(see Sect. 2.1) or the IP addresses have been spoofed. The specific E_i must be forwarded to the PE.

(ii) Let d_i, $m_i \in E_c$, given that i_i, $s_i \notin E_c$, then if $a_i \in R$ and $i_i = s_i$, the corresponding tuple $(H(F_i))$ shall be updated only when the message has been successfully authenticated. This registration message comes from a UE that has changed location and has been given a new IP address.

(iii) Let a_i, $m_i \in E_c$ given that i_i, s_i, $d_i \notin E_c$. If $a_i \in R$ then the corresponding registration message has been initiated from the same UE but the subscriber has changed. After the successful registration the corresponding s_c has to be updated with the incoming s_i. For instance, a user swaps the Universal Integrated Circuit Board (UICC) with another one utilizing the same UE and proceeds to a new registration procedure. The case where $a_i \notin R$ is covered in (i).

(iv) Let only s_i, $m_i \in E_c$, then there is application or network layer spoofing attempt and the corresponding message shall not be processed. The PE must be updated with the specific E_i.

(v) Let only i_i, $m_i \in E_c$, then it is straightforward that $s_{i,} \in E_c$, and thus, there is an application layer replay or SIP signaling attack and the message shall be dropped. The PE must be updated. The attacker has reused a previously gathered SIP message from another subscriber. The attacker's objective is to bypass authentication mechanisms.

(vi) Let the i_i, m_i, $s_i \in E_c$, the message includes an IMPI/IMPU from another subscriber. This identity theft attempt comes from an IA and the message shall be dropped. The E_i is forwarded to the PE. We can assume that the attacker is an insider because a registration for his UE already exists in the table (the only element that does not belongs to E_c is the d_i). This behavior may enable the attacker to charge the provided service to the actual IMPI/IMPU owner.

(vii) Let the m_i, i_i, s_i, $d_i \in E_c$, then also m_i, i_i, s_i, $d_i \in K_c$. Then, the sets K_i and K_c are identical $(K_i = K_c)$ and the message is a legitimate one and shall be processed. When $K_i = K_c$, the message is legitimate irrespectively if $a_i \in E_c$.

(viii) Let only the $i_i \in E_c$ then m_i, s_i, a_i, $d_i \notin E_c$. The message that corresponds to this specific tuple is spoofed and shall be dropped. The PE function must be updated.

(ix) Let the i_i, s_i, $d_i \in E_c$. If $a_i \in R$, then the corresponding message shall be processed because it can be considered as a legitimate one. The subscriber has initiated a registration procedure utilizing a new UE and the tuple has to be updated (with the new MAC address) after a successful authentication. If $a_i \notin R$, then the IPs of both protocols (SIP and IP) are spoofed or there is an ARP poisoning attempt. The correspondence between MAC and IP has changed and the sets $E_i \neq E_c$ and $K_i \neq K_c$.

(x) We know that $K_c \subset E_c$ because $a_c \notin K_c$ then of course neither $a_i \notin K_c$. Therefore if $K_i \cap K_c = \emptyset$, given that $E_i \cap E_c \neq \emptyset$, it is derived that only $a_i \in E_c$. Then, if the specific $a_i \in R$ and $i_i = s_i$ the message comes from a UE that has initiated a registration procedure for the first time. The tuple that corresponds to the specific E$_i$ has to be updated ($i_i \rightarrow i_c$ and $m_i \rightarrow m_c$) and stored to the registration table after a successful authentication. Otherwise, if $a_i \notin R$ or $i_i \neq s_i$ the message is spoofed while an unregistered UE tries illegally to forward a message or to be registered with forged IP.

(xi) Let only the $s_i \in E_c$ then m_i, i_i, a_i, $d_i \notin E_c$. The message is spoofed at the application layer. This can be a replay attempt or a SIP signaling attack initiated from an external user. The deduction that the attacker comes from the outside is derived from the fact that none of m_i, i_i belong to E_k; thus, the specific UE has not been registered until then. The PE is informed for that action.

If the cross-checking between E_i and E_c concludes that the message is not spoofed, then the E_i is forwarded to module II. Even though the specific message may not be spoofed, it is not known whether it is also a legitimate one (it may be part of a flooding attack).

4.3 Flooding attack detection method

The second module is fed with the E_i that has successfully passed the cross-correlation checking of module I. Every E_i has always its own same position on the table, and thus, the values are overwritten except for Init.D_Avg. The latter value is calculated during the initial handshakes of a specific UE with the server. If the T.Dist between two consequent messages is smaller than the average (T.Dist_Avg), the Trs value is incremented by 1. The T.Dist_Avg is calculated during a training period of 30 min in a normal traffic environment:

$$\text{T.Dist}_{\text{\_Avg}} = \sum_{i=1}^{m} \text{T.Dist}_i / z \tag{1}$$

where z is the number of Ci $\neq$ 0. An alarm is triggered (SSalarm) when the Trs exceeds a predefined number of messages that are below the T.Dist_Avg. The specific Ei tuple is send to the PE in order to deny access to the corresponding UE.

In case of distributed flooding attacks, the attacking entities may flood the server with low rate of consequent messages so as not to exceed the T.Dist_Avg value. Such a behavior can be detected utilizing the Init.D_Avg and Curr.D_Avg values in conjunction with the average of the incoming messages (Init.C_Avg) and the sum of them (Init.C_Sum). The Init.C_Avg and the Init.C_Sum can be measured during the training period by function (2) and (3) correspondingly:

$$\text{Init.C\_Avg} = \sum_{i=1}^{m} C_i / z \tag{2}$$

$$\text{Init.C\_Sum} = \sum_{i=1}^{m} C_i \tag{3}$$

$$\text{Init.C\_Sum\_Avg} = \sum_{k=1}^{m} \text{Init.C\_Sum}_k \tag{4}$$

where k is the number of executed trainings.

If function (4) grows with an unusual rate, an alarm is triggered (DSalarm1). A tolerance rate (tr) should be estimated according to the server's capabilities during high traffic periods. Thus, if the Curr.C_Sum > Init.C_Sum_Avg + tr, then a DS flooding attack is under way. As already mentioned, the attacking entities cannot be detected by calculating only T.Dist since it may not be exceeded and the alarm not triggered. Therefore, the suspicious subscribers are those with current value of C_i greater than Init.C_Avg + tr$_2$. Also attacks can be prevented by detecting variations between Init.D_Avg and Curr.D_Avg of every row. An alarm is triggered when the fraction in function (5) tends to a predefined number x.

$$\lim_{\text{Curr.D\_Avg} \to 0} \text{Curr.D\_Avg/Init.D\_Avg} = x. \tag{5}$$

Therefore, the decrement of the average response time of the specific tuple is calculated. If that happens, for instance, with a rate of 60 %, namely $x = 0.4$, during the first DS flooding alarm (DSalarm1), it can be derived that the Ei has been involved in the attack (DSalarm2), and thus, the PE must be informed in order to restrict its access to the server. Another attack case that can be detected by (5) is the increasing rate flooding attack [35].

In such situations, the attackers try to bypass a traffic rate-based detection mechanism by gradually increasing the attack rate. Therefore, the distance average is decreased gradually until very low values. This can be detected employing function (5) that calculates slight or enormous deviation in UEs' traffic behavior. The detection is illustrated in Fig. 10. The SSalarm can detect only CR attacks by calculating function (1), where the attackers send an huge amount of messages per time unit. On the other hand in IR attacks, the gradual increment of flooding rate slowly decreases the

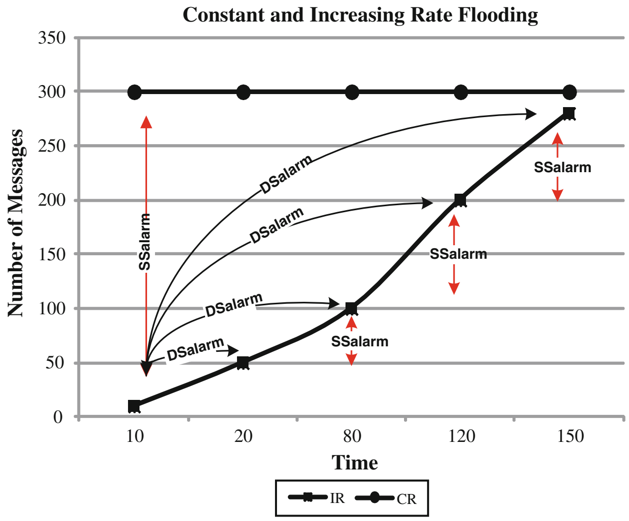

Fig. 10 Detection of increasing and constant rate attacks

T.Dist_Avg value, and thus, function (1) tends to be incapable to cope.

Another point of major importance is that function (5) provides the IDPS with the appropriate information for distinguishing between legitimate users and attackers when both trigger the DSalarm when the filter lacks adequate training. In such cases, the distributed source alarm is triggered, while the sum of C_i grows bigger over the time. The mechanism may trigger a false alarm for both legitimate messages and malicious ones but when it will seek for the specific C_i that have been involved in the attack, it will only detect the attackers. This occurs due to the fact that function (5) will still tend to 1 (note that Curr.C_Avg $\approx$ Init.C_Avg) for the legitimate users and to 0 for the actual attackers.

4.4 Evaluation

The proposed mechanism has been evaluated in terms of memory consumption and efficiency in attack detection. The test bed architecture (Fig. 11) consists of the Open IMS platform [36], three legitimate call generators (LCG), and one malicious call generator (MCG) for initiating flooding and spoofed message attacks. The IMS server with the proposed mechanism has been installed on a dual core machine at 2.4 GHz with 4 Gigabytes of RAM, while the call generators have been initiating messages through two 2 GHz dual core machines with 1 and 2 Gigabytes of RAM correspondingly. Finally, the attacker has been installed one a single core machine at 2.2 GHz with 1 Gigabyte of RAM.

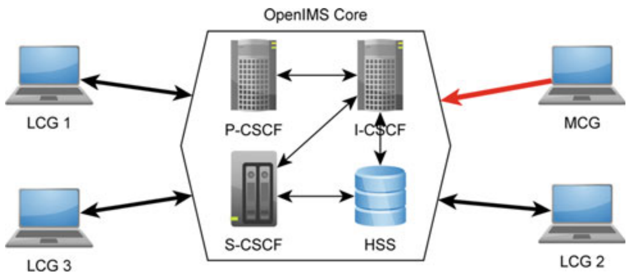

Fig. 11 The employed test bed architecture for evaluating the proposed mechanism

Table 3 The employed scenarios for evaluating memory consumption

CPS	Size	Description
20 cps	2k 5 k	In this scenario the Legitimate Call Generators (LCG) communicate through the IMS Core at a pace of 20 calls per second (cps). The bloom filter has a size of 2,000 and 5,000 cells
50 cps	2 k 5 k	In this scenario the (LCG) communicate through the IMS Core at a pace of 20 cps. The bloom filter has a size of 2,000 and 5,000 cells
70 cps	2 k 5 k	In this scenario the (LCG) communicate through the IMS Core at a pace of 20 cps. The bloom filter has a size of 2,000 and 5,000 cells

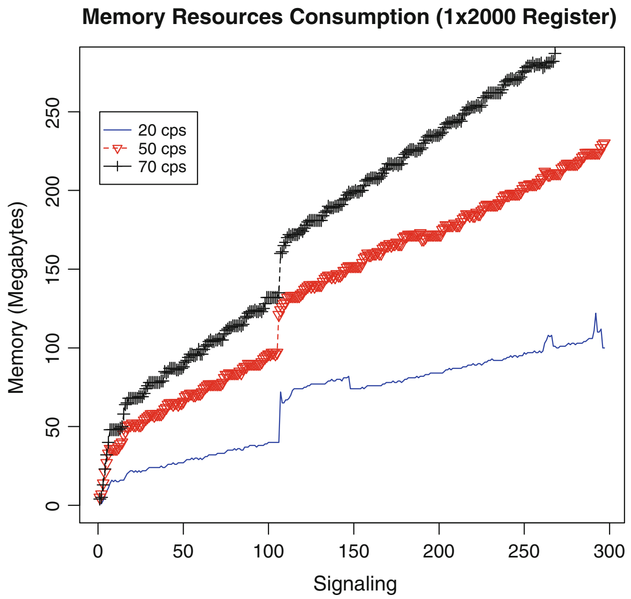

Fig. 12 Memory resources consumption in three different traffic scenarios with bloom filter size of 2,000 registers

4.4.1 Memory consumption

Three traffic scenarios have been employed for evaluating bloom's filter memory consumption with two different vector sizes per scenario. The scenarios are presented in Table 3. Considering Fig. 12, the allocated memory resources, when the bloom table employs 2,000 registers (which corresponds to 2,000 subscribers), is relatively low in all traffic scenarios. However, there is a difference of 140 % between the

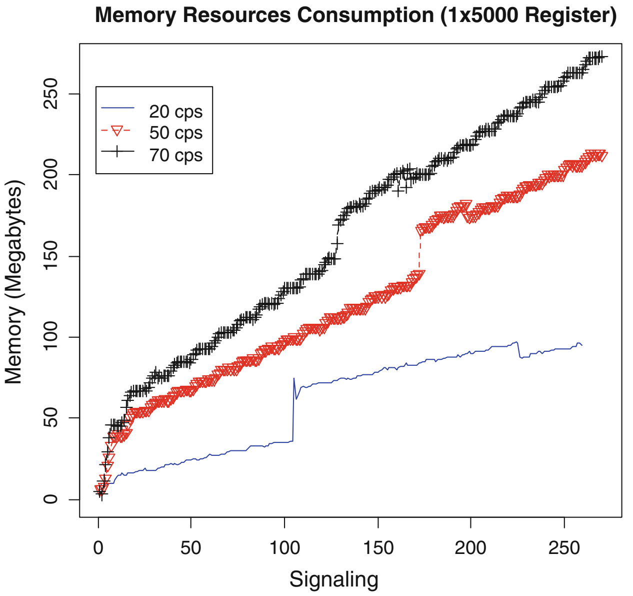

Fig. 13 Memory resources consumption in 3 different traffic scenarios with bloom filter size of 5,000 registers

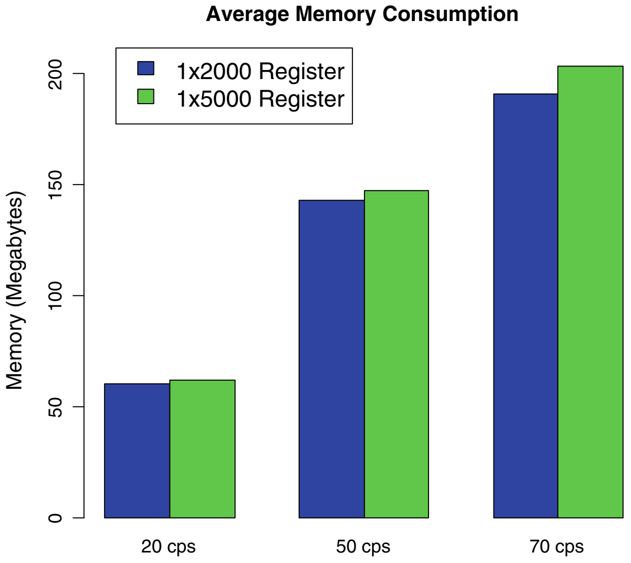

Fig. 14 Comparison of average memory consumption between different sizes of bloom filter

20 calls per second (cps) and 50 cps. On the other hand, a small increase (33 %) in memory consumption is observed when the traffic grows from 50 cps to 70cps. Almost the same results are derived when the bloom's size is increased to 5000 registers in order to host the corresponding amount of subscribers. There is a difference of 144 % from 20 to 50 cps and 38 % from 50 to 70 cps (Fig. 13). It is also illustrated that the different sizes of the bloom filter do not significantly affect the amount of the allocated memory resources as much as the increment of traffic does. The bar chart in Fig. 14 includes a comparison of the average memory consumption between the two different bloom sizes. In 20 cps scenario, both bloom sizes consume the same amount of memory while the same is true for the 50 cps scenario. Only a slight increment of 6 % can be observed in 70 cps scenario between the small and the larger bloom table.

4.4.2 Accuracy in attack detection

For an IDPS, it is extremely important to produce the lowest possible number of false positive alarms. A large number of false positives degrades the quality of the provided services while the system causes a denial of service to its own environment and consequently to the legitimate users. On the other hand, a mechanism that produces a large number of false negatives does not affect the legitimate users but it cannot efficiently detect attacks.

For visualizing the performance of the proposed mechanism, we use the receiver operating characteristics (ROC) graphs. To design such a graph, the following four values are required: the number of false positives (FP—Legitimate messages which are misclassified as attacks), the number of true positives (TP—Malicious messages which are detected as attacks), the number of false negatives (FN—Malicious messages that passed as legitimate ones), and finally the number of true negatives (TN—Legitimate messages that are correctly classified as legitimate traffic).

The true positive rate (TPR), also called *sensitivity* or *hit rate*, calculated by equation (6) [37].

$$\text{TPR} = \frac{\text{TP}}{\text{TotalPositives}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

The false positive rate (FPR), also called *false alarm rate*, calculated by equation (7).

$$\text{FPR} = \frac{\text{FP}}{\text{TotalNegatives}} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{7}$$

Legitimate and malicious traffic scenarios have been run on the same test bed architecture (Fig. 11) in order to calculate the TPs and FPs rates. All scenarios have been run with normal background traffic of 10 cps. The flooding and the spoofing detection modules are tested with three and five scenarios correspondingly (Table 4). Considering the ROC graph of Fig. 15, we can deduce that the spoofing module almost produces zero number of false alarms, and at the same time, it has detected all the malicious spoofed messages initiated by the MCG in scenarios S B2, S B3, and S B4.

False negative rates are slightly increased in increasing rate flooding attacks (Fig. 16) due to the fact that the specific attack requires many messages to develop. However, it does not have any impact to the system until it reaches the threshold values. At this point, the specific alarm is

Table 4 The employed scenarios for estimating mechanism's detection accuracy

Testing point	Sc. no.	Description
Module I	S B1	Legitimate Traffic between the LCG #1 and LCG #2 through the OpenIMS core at a pace of 10 cps
	S B2	MCG launches 1,500 messages with spoofed ID ("From" header) while S B1 is being carried out
	S B3	MCG launches 1,500 messages with spoofed MAC address while S B1 is being carried out
	S B4	MCG launches 1,500 messages with spoofed IP address while S B1 is being carried out
	S B5	LGC #3 launches 3,000 legitimate messages while S B1 is being carried out
Module II	S B6	MCG launches constant rate flooding attack when S B1 is being carried out
	S B7	MCG launches increasing rate flooding attack when S B1 is being carried out
	S B8	S B7 while the proposed IDPS mechanism is poorly trained and configured

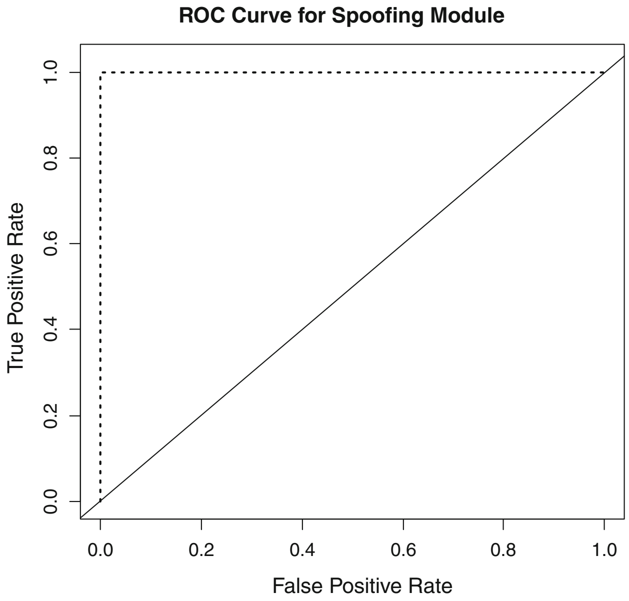

Fig. 15 Spoofing module's ROC graph for scenarios S B1–S B5. There are almost zero false alarms and the detection rate is very high

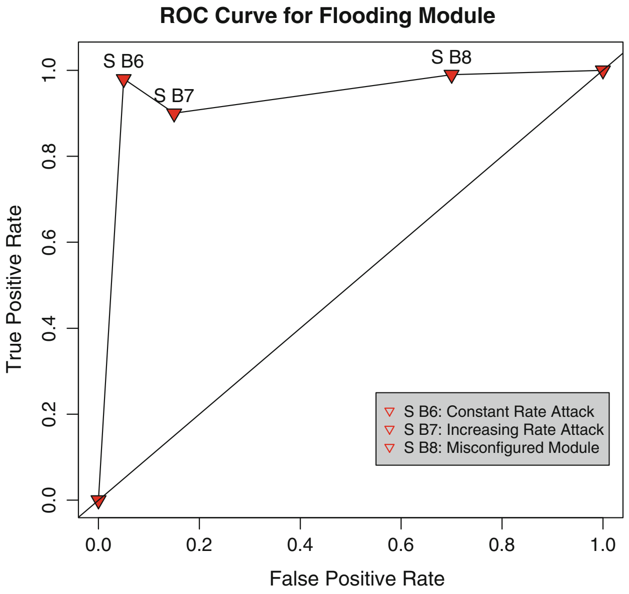

Fig. 16 Flooding module's ROC graph for scenarios S B6–S B8

triggered and the attacker is deterred. The same graph illustrates that the false positives tend to zero in constant rate attacks with a high detection rate (high TPR). On the other hand, considering S B8, we can pinpoint that the mechanism produces a large number of false positives and true positives at the same time. This is due to the inadequate training of the mechanism. This occurs when the scenarios exceed the time of the training period (30 min), and thus, the collected mean values, as described in Sect. 4.3, are much smaller. Therefore, only the distributed source alarm is triggered, while $\sum Ci$ grows bigger over time. However, the mechanism may wrongly trigger the DSalrm for both legitimate messages and malicious ones but when it will seek for the specific C_i that have been involved in the attack it will only detect the attackers. This happened due to the fact that the function (6) tends to 1 for the legitimate users (while the Curr.C_Avg ≈ Init.C_Avg) and to 0 for the actual attackers.

5 A comparative study

Various different approaches and proposals of how VoIP and IMS environments can be hardened against attacks can be found in the related literature. However, most of them are focused on the detection without being able to actively deter the attackers, while there are only a few that can discourage specific cases of the above-mentioned attacks.

In [14], a flooding detection model is introduced based on priority queues. More specifically, it deploys two queues, one of high and one of low priority. All the INVITE messages are inserted in the low-priority queue, while the responses are inserted in the high priority one. The messages in low-priority queue will be processed only when the high priority queue is empty. The result is that the legitimate requests are the ones handled first while their responses are in higher priority. On the other hand the INVITE messages are processed with an increased delay but the server can still be online avoiding

DoS consequences. Moreover, the illegitimate INVITEs are discarded faster since they do not come with responses and thus the high priority queue remains empty. Nevertheless, this model does not actively deter or block the attacker but only mitigates the effects of the flooding attack. Furthermore, the attacker can bypass the mechanism by flooding the server and consequently the two queues, both with INVITE requests and responses (e.g. 100, 200, 180 etc.).

In [13], the authors utilize bloom filters and a SIP-specific metric called "session distance", in order to correlate the number of the received requests with the number of 200 OKs and ACKs and thus detect deviations from the normal traffic. Similarly, in [12] a DoS mechanism that extends the detection to the transport layer is proposed. The mechanism detects traffic abnormalities by calculating the Hellinger distance between requests and responses taking into account not only the SIP traffic but also the transport protocol's traffic. Both of the mechanisms do not provide a methodology to detect and deter the attackers but are only triggering alarms when a flooding attack is under way. Moreover, they are only focused on the INVITE request flooding attack case.

Faweett [38] describes a mechanism that monitors REGISTER messages. A successful registration takes place when a 200 OK is received by the server which includes the REGISTER value in the CSEQ header. Then, a tuple is created in a table containing information gathered from the SIP message: the user's identity (UID), the UE's IP address a timestamp and the duration of the specific registration in seconds as it is derived from the expires header. The UID acts as primary key in the table. This table is actually a white list and therefore only the stored UIDs will be processed. This mechanism does not offer any protection against IAs which have a legitimate subscription and can authenticate all of their requests. Moreover, it is unable to deter INVITE flooding attacks and it cannot be deployed in IMS infrastructures in conjunction with security tunnels because the messages do not contain SIP authentication except for the first REGISTER request. Messages with forged UIDs can bypass the mechanism, and finally, DDoS attacks are not handled.

Wu [10] presents a mechanism which utilizes rules in order to examine events developed in VoIP architectures. An alarm is triggered when a collection of events corresponds to a predefined rule. This alarm indicates that an attack occurs. The majority of the rules are based on an end-to-end matching rule which actually detects deviations in signaling flows. This mechanism is able to detect MitM and billing fraud attacks. However, an attacker may bypass the rules by launching a layer 3 impersonation attack. Finally, this approach does not offer protection against many MitM, signaling, and flooding attacks.

A flooding attack prevention mechanism is presented in [39]. The authors propose a detection mechanism that is able to detect DoS including some SIP signaling attacks. Specifically, Honey Pot architecture is deployed in order to provoke attacker's interest and thus gather useful data toward the detection of such attacks. Utilizing anomaly and signature-based detection techniques, the mechanism creates profiles of "normal behavior" for users and network entities and signatures of known attacks. Any deviation from the normal behavior standards can be considered as an attack. An attack is detected by correlating different events through specific rules. For instance, the BYE signaling attack can be detected by spotting orphan RTP flows after a period of time (only from one participant while the other has received the BYE message) utilizing signature-based correlation with attack patterns.

In [40], the authors propose a mechanism for integrity protection in SIP messages. Specifically, all the contents of the SIP messages are hashed including the header and the body of the message with the user's password. Also, this model proposes the embodiment of an extra header (Verify-Body) which contains all the necessary information (e.g., username, hash function, realm, etc.) for the server in order to verify the integrity of the message. Although this mechanism can deter signaling attacks, replay, and MitM attempts in the SIP layer, it cannot prevent flooding attacks and MitM in all remaining layers.

Another approach for detecting spoofed calls is presented in [41]. The authors propose a method for profiling networks by extracting characteristics from audio streams. The extracted data are a combination of noise characteristics and packet loss which are adequate to build call/provider profiles. Their proposal is based on the observation that while the audio stream traverses through different networks, it is re-encoded by each network using different codes and bit rates (different network technologies use different audio codec specifications). As the audio is re-encoded, its quality is degraded providing enough information for the mechanism to make fingerprints from these artifacts. This approach can efficiently detect the actual call origin but only when the audio stream is included. Thus, all the signaling attacks cannot be deterred. Moreover, it cannot detect spoofing attacks when the attacker comes from the same physical location, and finally, it does not address flooding attacks.

A comparison among the above-mentioned proposals, with respect to their efficiency in detecting and preventing the attacks described in Sect. 2, is provided by Table 5.

6 Conclusions and future work

This paper introduces an intrusion detection and prevention system (IDPS) that can be deployed both in IMS and VoIP infrastructures that utilize the SIP as the signaling protocol. The detection covers most of the spoofed SIP message attacks and also flooding attacks when originated either from

Table 5 Comparison of security mechanisms

Layer	Threat category	Attack	Proposed		[14]		[10]		[39]		[12,13]		[38]		[40]		[41]	
			D	**P**	**D**	**P**	**D**	**P**	**D**	**P**	**D**	**P**	**D**	**P**	**D**	**P**	**D**	**P**
Layer 5	SIP signaling	BYE	✓	✓	X	X	✓	X	✓	✓	X	X	✓	✓	✓	✓	X	X
		CANCEL	✓	✓	X	X	✓	X	✓	✓	X	X	✓	✓	✓	✓	X	X
		Re-INVITE	✓	✓	X	X	✓	X	X	X	X	X	✓	✓	✓	✓	X	X
		UPDATE	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓	X	X
	Masquerade/ID Theft	UE impersonation (SIP-IPs)	✓	✓	X	X	X	X	X	X	X	X	✓	✓	X	X	X	X
		User impersonation (SIP IDs)	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓	✓	✓
	MitM	Registration expiration	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓	X	X
		Bid down	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓	X	X
		Generic authentication	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓	X	X
		Conference interception	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓	X	X
	Replay	SIP replay	✓	✓	X	X	X	X	X	X	X	X	✓	✓	✓	✓	X	X
	Flooding attacks	Non-INVITE	✓	✓	X	X	X	X	X	X	X	X	X	X	X	X	X	X
		Invite	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	X	X	X	X	X
		Single source	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	X	X	X	X	X
		Distributed source	✓	✓	✓	✓	X	X	✓	✓	✓	X	X	X	X	X	X	X
Layer 3	Masquerade (L3)	UE IP spoof	✓	✓	X	X	X	X	X	X	X	X	X	X	X	X	✓	✓
Layer 2	MitM (L2)	ARP poison	✓	✓	X	X	X	X	X	X	X	X	X	X	X	X	X	X

With **D** is denoted that the mechanism provides only attack detection while with **P** that it can also prevent the attack

single or distributed sources. Its design poses a lightweight mechanism, free of complex and resource demanding calculations, a very crucial factor in such environments where QoS is of top priority. It also provides high detection rates with false alarms that approach zero (even when the filter is inadequately trained and configured), a parameter that is very crucial for such mechanisms. The online position in the network offers real-time detection and prevention of the attacks, and its cross-layer structure facilitates the detection of an incident from the lower layers of the internet protocol stack. It is also covers the majority of the security breaches that can be developed in SIP environments in relation to the ones proposed in literature [10,12–14,38–40].

Our objective is to extend the mechanism in order to cover the rest of the attacks that can be launched against such infrastructures. Specifically, malformed message attacks are posing a threat even in the more contemporary implementation and can cause DoS introducing large amount of overhead in the communication [42].

References

1. 3GPP: TS 23.228: IP Multimedia Subsystems (IMS). Third Generation Partnership Project, Technical Specification Group Services and System Aspects (2011)
2. Rosenberg, J., et al.: RFC 3261: SIP: Session Initiation Protocol (2002)
3. Tanase, M.: IP spoofing: an introduction. Secur. Focus **11** (2003). Available at:http://www.securityfocus.com/infocus/1674
4. Wagner, R.: Address resolution protocol spoofing and man-in-the-middle attacks. The SANS Institute (2001). Available at:http://rr.sans.org/threats/address.php
5. Geneiatakis, D., et al.: Survey of security vulnerabilities in session initiation protocol. IEEE Commun. Surv. Tutor. **8**, 68–81 (2006)
6. Park, Y., Park, T.: A survey of security threats on 4G networks. In: IEEE Globecom Workshops, Washington, DC, pp. 1–6 (2007)
7. Keromytis, A.: A survey of voice over IP security research. In: Prakash, A., Sen Gupta, I. (eds.) Information Systems Security, vol. 5905, pp. 1–17. Springer, Berlin (2009)
8. 3GPP: TS 33.203: 3G security; Access security for IP-based services (Release 10). Third Generation Partnership Project, Technical Specification Group Services and System Aspects (2010)
9. Franks, J., et al.: RFC 2617: HTTP authentication: basic and digest access authentication. Internet Eng. Task Force (1999). Available at:http://www.ietf.org/rfc/rfc2617.txt
10. Wu, Y., et al.: Intrusion detection in voice over IP environments. Int. J. Inf. Secur. **8**, 153–172 (2009)
11. Wu, Y., et al.: Scidive: A stateful and cross protocol intrusion detection architecture for voice-over-ip environments. In: Proceedgins of the 2004 International Conference on Dependable Systems and Networks (DSN 2004), Firenze, Italy, pp. 433–442 (2004)
12. Sengar, H., et al.: Detecting VoIP floods using the Hellinger distance. IEEE Trans. Parallel Distrib. Syst. 794–805 (2008)
13. Geneiatakis, D., et al.: Utilizing bloom filters for detecting flooding attacks against SIP based services. Compu. Secur. **28**, 578–591 (2009)
14. Wan, X.Y., et al.: A SIP DoS flooding attack defense mechanism based on priority class queue. In: IEEE International Conference on

Wireless Communications, Networking and Information Security (WCNIS), Beijing, China, 25–27 June, pp. 428–431 (2010)
15. Srinivasan, R., et al.: Authentication of Signaling in VoIP Applications. In: Asia-Pacific Conference on, Communications. pp. 530–533 (2005)
16. Argyroudis, P.G., et al.: Performance analysis of cryptographic protocols on handheld devices. In: Third IEEE International Symposium on Network Computing and Applications (NCA 2004), pp. 169–174 (2004)
17. Shen, C., et al.: The impact of TLS on SIP server performance. In: IPTComm 2010: 4th Conference on Principles, Systems and Applications of IP Telecommunications Principles, Systems and Applications of IP Telecommunications, Munich, pp. 59–70 (2010)
18. Geneiatakis, D., et al.: SIP Security Mechanisms: A state-of-the-art review. In: Proceedings of Fifth International Network Conference, Samos, Greece, pp. 147–155 (2005)
19. Geneiatakis, D., et al.: SIP message tampering: the SQL code injection attack. In: Proceedings of 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2005), Split, Croatia (2005)
20. Bremler-Barr, A., et al.: Unregister attacks in SIP. In: 2nd Workshop on Secure Network Protocols, NPSec, pp. 32–37 (2006)
21. Abdelnur, H., et al.: Abusing SIP authentication. In: ISIAS' 08: Fourth International Conference on Information Assurance and Security, pp. 237–242 (2008)
22. Klein, A.: BIND 9 DNS cache poisoning. Available:http://www.trusteer.com/docs/bind9dns.html (2007)
23. Vrakas, N., et al.: A call conference room interception attack and its detection. In: Presented at the 7th International Conference on Trust, Privacy and Security in Digital Business, Bilbao, Spain, (2010)
24. Asokan, N., et al.: Man-in-the-middle in tunnelled authentication protocols. Lecture Notes in Computer Science, vol. 3364, p. 28 (2005)
25. Xia, H., Brustoloni, J.: Hardening web browsers against man -in-the-middle and eavesdropping attacks. In: Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, pp. 498–498 (2005)
26. Zhang, R., et al.: On the feasibility of launching the man -in-the-middle attacks on VoIP from remote attackers. In: Presented at the 4th ACM Symposium on Information, Computer and Communications Security, Sydney, Australia, March (2009)
27. Callegari, C., et al.: A novel method for detecting attacks towards the SIP protocol. In: International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), pp. 268–273 (2009)
28. Sisalem, D., et al.: SIP Security: Wiley (2009)
29. Vrakas, N., Lambrinoudakis, C.: A cross layer spoofing detection mechanism for multimedia communication services. Int. J. Inf. Technol. Syst. Approach (IJITSA) **4**, 32–47 (2011)
30. Postel, J.: RFC 793: TCP: transmission Control Protocol. (1980)
31. Bellovin, S.: Security problems in the TCP/IP protocol suite. ACM SIGCOMM Comput. Commun. Rev. **19**, 48 (1989)
32. 3GPP: TR 33.978 Security aspects of early IP Multimedia Subsystem (IMS). Third Generation Partnership Project, Technical Specification Group Services and System Aspects (2008)
33. ETSI: TS 187.003: Telecommunications and internet converged services and protocols for advanced networking (TISPAN): Security Architecture. (2008)
34. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM **13**, 422–426 (1970)
35. Udhayan, J., Hamsapriya, T.: Statistical segregation method to minimize the false detections during DDoS attacks. Int. J. Netw. Secur. **13**, 152–160 (2011)
36. OpenIMS: Fraunhofer Fokus. Available: http://www.openimscore.org
37. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. **27**, 861–874 (2006)
38. Chen, E.Y., Itoh, M.: A whitelist approach to protect SIP servers from flooding attacks. In: IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), Vancouver, BC, 8–10 June, pp. 1–6 (2010)
39. Nassar, M., Niccolini, S.: Holistic VoIP intrusion detection and prevention system. In: Bond, G.W., Schulzrinne, H., Sisalem, D. (eds.) Principles, Systems andApplications of IP Telecommunications (IPTComm 2007). New York, USA, pp. 1–9 (2007)
40. Takahara, H., Nakamura, M.: Enhancement of SIP Ssgnaling for integrity verification. In: 10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT), pp. 289–292 (2010)
41. Balasubramaniyan, V.A., et al.: PinDr0p: using single-ended audio features to determine call provenance. In: Proceedings of the ACM Conference on Computer and Communications Security (CCS), pp. 109–120 (2010)
42. Vrakas, N., et al.: IS IP MULTIMEDIA SUBSYSTEM AFFECTED BY 'MALFORMED MESSAGE' ATTACKS? An Evaluation of OpenIMS. In: SECRYPT 2011, the International Joint Conference on e-Business and Telecommunications, Seville, Spain, 18–21 July (2011)

Author Biographies

Nikos Vrakas is currently a Ph.D. candidate at the University of Piraeus. He has received his bachelor degree in Mathematics at the University of the Aegean and the M.Sc. in Information and Communication Security where he received scholarship for his efforts. He has authored papers as well as a chapter in a book in the field of IT Security. His research interests include availability, intrusion detection, and privacy preserving mechanisms of the Session Initiation Protocol (SIP) in IP Multimedia Subsystem (IMS) of the Next Generation Networks (NGNs).

Costas Lambrinoudakis holds a B.Sc. (Electrical and Electronic Engineering) from the University of Salford (1985), an M.Sc. (Control Systems) from the University of London (Imperial College -1986), and a Ph.D. (Computer Science) from the University of London (Queen Mary and Westfield College - 1991). Currently, he is an Assistant Professor at the Department of Digital Systems, University of Piraeus, Greece. From 1998 until 2009, he has held teaching position with the University of the Aegean, Department of Information and Communication Systems Engineering, Greece. His current research interests are in the areas of Information and Communication Systems Security and of Privacy Enhancing Technologies. He is an author of more than 85 scientific publications in refereed international journals, books and conferences, most of them on ICT security and privacy protection issues. He has served as program committee chair of 5 international scientific conferences and as a member on the program and organizing committees in more than 60 others. Also he participates in the editorial board of two international scientific journals and he acts as a reviewer for more than 30 journals. He has been involved in many national and EU funded R&D projects in the area of Information and Communication Systems Security. He is a member of the ACM and the IEEE.