Project no.:	**COOP-32720**
Project full title:	**SPam over Internet telephony Detection sERvice**
Project Acronym:	**SPIDER**
Deliverable no.:	**D2.2**
Title of the deliverable:	**Spit detection and handling strategies for VoIP infrastructures**

Contractual Date of Delivery to the CEC:	31.03.2007
Actual Date of Delivery to the CEC:	
Author(s):	G. F. Marias (AUEB), S. Dritsas (AUEB), M. Theoharidou (AUEB), J. Mallios (AUEB), L. Mitrou (AU), D. Gritzalis (AUEB), T. Dagiuklas, (AU), Y. Rebahi (Fokus), S. Ehlert (Fokus), B. Pannier (Eleven), O. Capsada (VozTelecom), J. F. Juell (Telio)
Participant(s):	Fokus, AU/AUEB, VozTelecom, Telio, Eleven
Work package contributing to the deliverable:	WP2
Dissemination level:	Public
Version:	01
Total number of pages:	40

Abstract:

SIP provides new means for the establishing and maintaining multimedia, conference and voice sessions, as well as exchanging instance messages and presence information. On the other hand it suffices from several vulnerabilities that could be exploited by potential threats to contact SPam over Internet Telephony (SPIT) attacks, which generality are defined as the transmission of bulk unsolicited messages and calls. In this deliverable we have conducted a survey of the anti-spit mechanisms and frameworks that have been designed and implemented so far, followed by a theoretical evaluation framework, which is based on qualitative and quantitative criteria in terms of effectiveness. We have identified attack scenarios exploiting threats and vulnerabilities in order to understand how the attacks are likely to be delivered, in which portions of the network and in which phase of the call establishment. Additionally, we examine which of the SPIT identification criteria each framework fulfils. This document is intended to provide the basic requirements that should be encountered when a preventive, detecting and handling SPIT mechanism is under design and specification, as well as to identify the relative study contacted so far in this research and development area.

Keyword list: VoIP, SIP, Spit, anti-spit, Prevention, Detection, Handling, State of the art, assessment criteria, theoretical evaluation.

Table of contents

1 Introduction

The spam over Internet telephony (spit) problem demonstrates several similarities with the e-mail spam (hereafter SPAM). Even if spit statistics are not yet available in high volumes, spit attack patterns might reproduce those of spam. Eventually, anti-spam mechanisms might prove useful in the VoIP domain as a feasible solution for spit mitigation. To discuss such applicability, the requirements enforced by the Session Initiation Protocol (SIP) when dealing with spit should be defined. In the first part of this deliverable we deal with these issues, defining the spit problem space more concretely, surveying if some of the e-mail anti-spam solutions might serve as a base for enhanced anti-spit mechanisms.

In the second part, we discuss existing implementations or proposed protocols and architectures that can be found in the SIP domain. These provide for the initial definition and recommendation of anti-spit categories, namely prevention, detection, and handling spit policies. We discuss legal issues associated with the existing proposal, as well as their basic concepts.

Additionally, we deal with requirements for spit avoidance, situated by SIP services providers, users and technology providers. Such technical requirements are classified as qualitative and quantitative criteria that should be considered when fighting spit. For that reason we record how the existing implementations or proposed solutions respond to each one of the requirements, and evaluate each proposal based on these criteria.

Finally, we seek for identifying criteria that are not fulfilled, making the corresponding objective decisions for the design of an innovative solution in the next phase of the project.

2 Mail Anti-spam solutions on the SPIT domain

2.1 Common profiles of SPAM attacks

VoIP Spamming tools are various, ranging from simple gathering of SIP valid URIs tools to more sophisticated spreading calls and messages tools. However, they can be summarized as follows,

- *Harversting:* this means building a database of targets by finding valid SIP URIs. There are different ways to collect thousands of SIP URIs on the Internet. When exchanging SIP messages, if some adequate scripts are in place, they can look up in the headers of the SIP messages and save the information related to the ”*From”* and *”To”* fields. Similarly to email, another technique for harvesting is to search, using appropriate engines, into Web pages for SIP URIs. These programs can check for instance for information such as *”contact: link”*.
- *Dictionary attacks:* This consists of generating SIP URIs using a predefined list of words at a particular domain and sending SIP requests to them. Those that bounce back are purged, those that do not are assumed to be active and added to the spammer list. This technique works fine in case of servers with huge numbers of subscribers. One can start with *”aaaaa@example.com”*, then *”aaaab@example.com”* and so on.
- *Open proxies:* An open proxy is a SIP server opened to the world for almost any SIP request, allowing anybody to remain anonymous while crawling the net. These proxies are very useful for spammers as they allow them to stay anonymous while sending unsolicited messages
- *Open relays:* An open relay is a proxy that accepts third party relays of SIP messages even they are not destined to its domain. These relays are used by spammers to route large volumes of unwanted SIP messages
- *Compromised hosts:* Hosts, which are infected with several types of viruses and/or spyware, are used today to distribute Spam’s. A host gets infected due to a user who opens an infected email attachment or visits a compromised web server, which are using browser exploits to transfer the virus to the host. The virus is getting the receptions and the content of a Spam from several root Spam Servers. With this technique the amount of Spam’s has more than six fold last year [14].

2.2 Email Anti SPAM systems Description

In this section, we will discuss some techniques that can be used for dealing with SIP spam. Yet, a first class of solutions adopted from email spam detection can be considered. This class is represented in particular by black lists, white lists and content filtering. As these techniques are not in general effective and cannot deal with all the different spam forms, combining these techniques together or using them within more complete solutions seems to be mandatory.

The SIP protocol uses textual messages to establish a communication between end users. A SIP message generally comprises two parts, the header and the body. In the call form, the message (for instance the SIP INVITE message) involves a header and an SDP part. However, in the IM form, the message, namely the SIP MESSAGE message, comprises a header and some text that refers to the message content.

To be able to classify a SIP message as spam, one can analyze the header of the message, the body of the message or both parts of the message. The message header can provide information such as, the sender is known as a spammer, the sender is providing misleading header information, the sender is sending the request to a large number of users. The content of the message can reveal information related to various topics: software sales, insurance offers, porn related files, semi-legal drugs products, etc. Classifying messages by filtering their headers is certainly the easiest way to deal with SIP spam as we only need to check whether the address of the sender belongs to a non desirable list or to a favorite one. The subject field (if it is used) of the SIP message can also be analyzed and used for blocking spam messages. Notice that most of the email spam detection mechanisms can be applied to the IM spam form because of the textual form similarities, however, some of these mechanisms might not be applicable to the call spam form. For instance, email content filtering tools could be adopted to deal with the IM spam, however, filtering the SDP part of an SIP INVITE message might not be helpful. In addition, filtering the relayed stream content, even with a very sophisticated speech recognition tool does not bring that much as the message has already reached the recipient.

The next sections will discuss some technical ways to reduce SIP spam. Each technique will be briefly presented and the focus will be more on its cons and pros.

2.2.1 Prevention-based filtering

Preventive here means dealing with spam even before it leaves the sender's domain. Every SIP provider can collaborate in reducing spam by blocking spam messages that are issued from his domain before they are sent out. In fact, a SIP provider is assumed to provide services for users, not to check what the users achieve through these services. However, as spam is a general problem, if a SIP provider does not collaborate, even not so much, in blocking spam messages, he will probably be the target of spammers and in this case, he will have to update his infrastructure with anti-spam systems which will be very expensive.

Among the techniques that a SIP provider can adapt to block spam messages issued from his domain is to check whether the used IP addresses are not fake. Indeed, IP spoofing is a mechanism that allows spammers to hide their identities and to be unreachable. The SIP provider can before sending out any SIP request, checks whether the used sending IP addresses are legitimate. In the opposite case, some measures will be taken against the senders. Another scheme that could be also used by SIP providers is signing every outgoing SIP message and provide a way for the recipient SIP providers to check the signature.

2.2.2 White lists

A white list is a set of senders' SIP addresses (or identifiers) that a user accepts calls or IMs from. These lists can be found under the names: address book, contact list, phone books, and are accessed and maintained in general by the users that the lists belong to. The information involved in a white list is gathered either by allowing the user, through an interface, to set the list by adding users SIP addresses and deleting others or by allowing the outbound SIP proxy to keep track of the *"To"* field information within the SIP messages sent out by the white list owner. In the latter situation, the access to this list can also be given to the user in order to let him setting his preferences in a better way, i.e., if user X has contacted user Y and does not want him to be on his white list, user X can delete the corresponding information from what the outbound proxy has collected.

Even using white lists is efficient for blocking spam messages, this technique, if it is used alone, is very restrictive and not realistic as communications are established only with the

users on the white lists. You might need, occasionally, to give your phone number (or SIP address) to a person that is assumed to provide you with a certain service such as repairing your car or keeping you updated with your bank offers. It may happen that one of your old friends wishes to hear from you. If these persons are not on your white lists, they will never be able to contact you.

A variation of the white lists technique consists of challenging each message that is issued by a user, which is not on the white list. Note that this variation can be applied to both IM spam and call spam forms. In the former case, no significant changes are required on the SIP server side. However, in the call spam case, when the SIP server receives a request, it is redirected to an Interactive Voice Response (IVR) service that plays a short audio file and asks for some DTMF input. In general, the way of achieving the challenge/response operations is described in section…. In this case, spam messages generated by machines will be systematically rejected as these machines are in general not capable of providing correct answers to the sent challenges. Unfortunately, this variation consumes more bandwidth on the network. And also language understanding problems could occur because the IVR could not predict which language the caller might be able to speak or not.

2.2.3 Black lists

Black (or block) lists are sets of "bad" senders. The information maintained in these lists are the IP addresses of the domains or the users that we consider as spammers. The messages received from the spammers will be rejected or uploaded to a "quarantine" bucket for further review. In fact, when a SIP request is received, we look up the domain name in the DNS black list repository and if the query is successful, the SIP request will be rejected. Black lists can reside either on the client or server side, however, the second option is more convenient as these lists need to be updated constantly.

Although, black lists can provide a reasonable barrier to the SIP spam, they have to be used with caution. In fact, the identity claimed by a spammer as well as the claimed domain might be some fake data or represent some information related to another user (or domain) who has nothing to do with spam. One of the most difficult problems in dealing with spam is the inability to determine accurately who sent the message and how it was routed in the Internet. Without a strong authentication mechanism, these lists can harm more than help in spam detection especially that entire domains may be blocked even they are not guilty. Enhancing black lists with domain verification will increase their efficiency.

2.2.4 Content filtering

The SIP standard uses textual messages to establish a communication between end users. A SIP message generally comprises two parts, the header and the body. In the call form, the message (for instance the SIP INVITE message) involves a header and an SDP part. However, in the IM form, the message, namely the SIP MESSAGE message, comprises a header and some text that refers to the message content.

To be able to classify a SIP message as spam, one can analyze the header of the message, the body of the message or both parts of the message. The message header can provide information such as, the sender is known as a spammer, the sender is providing misleading header information, the sender is sending the request to a large number of users. The content of the message can reveal information related to various topics: software sales, insurance offers, porn related files, semi-legal drugs products, etc. Classifying messages by filtering

their headers is certainly the easiest way to deal with SIP spam as we only need to check whether the address of the sender belongs to a non desirable list or not. The subject field (if it is used) of the SIP message can also be analyzed and used for blocking spam messages. Notice that most of the email spam detection mechanisms can be applied to the IM spam form because of the textual form similarities, however, some of these mechanisms might not be applicable to the call spam form. For instance, email content filtering tools could be adopted to deal with the IM spam, however, filtering the SDP part of an SIP INVITE message might not be helpful. In addition, filtering the relayed stream content speech recognition tool does not bring that much because first even if the recognition is very sophisticated, it is very easy for the Spammer to obscure the voice message with at example other tones, music and/or rustling. No computer based speech recognition system is able to handle this by today. The same problems do we see today in Email Spam's with Images, which contains the Text. The second reason for why speech recognition systems are not very helpful is that the message has already reached the recipient.

In general, content based spam filters analyze the text of the message and provide back how often some specified spam key words have occurred. According to this analysis, a score is assigned to the received SIP message. After crossing a predefined threshold, the SIP message is declared as spam. The problem with this kind of filters is that the system needs to be constantly administrated. It needs somebody who adds new phrases and rules, which are able to detect new kinds of Spam's. The other problem is that the risk of false positives with this kind of systems is very high. Only if a text contains the words "Viagra" and "Sex" at example does not mean instantly that is a SPIT or Spam.

As in commercial spam, the purpose of the message is to cause the receiver to buy a product or to use some services offered by the spammer, the filter can use the URIs, the phone numbers and email addresses that the receiver is assumed to use in order to contact this spammer as spam keywords or selected parts that could be hashed by the filter as mentioned earlier. The spammers may send spam messages with random input to get around the filter, however, the personal information that spammers provide to be contact cannot change a lot.

But the Spammers can try to hide these information's and as we see in Email Spam messages today their are creative to do so. So a telephone number at example could be written as +49-30-520056-150, but also as + 49_3O.520O56,I50 or much worse each digit will be given as written words. Each human is able to read and understand it, but to write an algorithm which is able to identify this example as the given number is not that easy. Even with URIs a lot of obscure things are possible as we see today in the Email Spam market.

Calculating hashes of some selected parts of each message which are stored in a database is an other approach. Identified Spam's or SPIT's will be marked as Spam in the database so each message within the same Spam wave could be identified as Spam. The reason of storing hashes of some selected parts of the spam messages instead of the entire spam messages is to keep a reasonable storage capacity. When a new message is received, the filter checks the hashes of the selected parts of the received message against the hashes in the database. The storage capacity could be lowered if a algorithm will be found which calculates this hash in a way that is able to count the amount of SPITs with the same hash. Hashes with the count of 2 or less could be removed from the database very early. That's because the goal of Spammers is always to reach as much as possible potential customers. So only hashes of active SPIT waves will stay in the database. Old hashes or hashes with low counters will be removed. The important things on hashes are which parts of a message will be used to calculate the hash and how robust is the hash. Robust means in this case what needs the Spammer to change within a message to result the hash calculation in two different hashes. If the algorithm is able to

recognize two slightly changed messages as the same and if it does not recognize two complete different messages as the same the algorithm is so called robust.

These hash technique works for IM spam but maybe even on SPIT messages if the algorithm is able to deal with binary or much better with audio stream data. The problem is that if the stream is already established the SPIT reaches the user what we are trying to prevent. As we see today with Spam messages in the traditional PSTN the Spammers are able to transfer their messages completely to answering machines. They start to play their message after the "beep" which means they are have something like a speech or "beep" detection in their software. So the idea to filter also SPIT's with a hash technology is to simulate an answering machine to get also parts of the SPIT message stream. If the Spammer sends its message after the beep or even earlier the first seconds of the audio stream will be feed to the hash algorithm which is able to compare it to already identified SPIT's. From the human caller perspective the same problems as with the Challenge/Response technique described later on can occur. The message must be clear to understand, the language boundaries and more must be thought carefully. But even if the simulated answering text was not understood from the caller or the caller starts to speak because he does not recognize that he is talking to an answering machine the hash does not results in a as Spam marked hash and the caller will be transferred to the designated phone.

The content filtering techniques are interesting in the sense that they do not require any changes on the current SIP clients or servers. Unfortunately, these techniques cannot guarantee 100% accuracy of spam because of the huge number of spam messages types and contents. To cope better with this problem, more spam information needs to be stored, this will, however, require more storage capacity. On the other side, let us note that this technique can be applied better to IM spam because in the call spam case, filtering the relayed stream content means that the call has already reached the recipient or more complicated techniques like the simulated answering machine must talk place. Another alternative consists of filtering the SDP part and verify whether the involved information already exists in the spam repository.

2.2.5 Challenge/response technique

When this technique is used, a challenge is sent to the sender of the message and the latter is processed if and only if the sender answers to the challenge correctly. As machines generate most of spam, this technique can be helpful. The challenges should be easy and various, starting, for instance, from analyzing a picture (in the case of IM spam) to answering to a recorded audio file such as "which day is today?" or "press the keys 1-2-3" in the case of call spam. In the latter, an Interactive Voice Response (IVR) system can be used. Thus, when a SIP request is received, it is redirected to the IVR system that plays the audio file and sets the challenge. If this response is correct, the SIP request is processed.

To be efficient, this technique needs to use various challenges forms and update them regularly. This will present a continuous challenge for the spammers. The challenge of building this technique is to provide a system, which makes sure that many as possible human callers are able to answer the challenge. So language boundaries, disabilities and other barriers should be considered while selecting the right questions. On the other side, non-spam users will probably feel annoyed if they need to respond to a challenge whenever a communication is needed. To overcome this problem, the challenge/response technique can be combined with some other spam detection schemes, for instance white lists that allow a

first classification of spam messages. In this case, only messages from unknown senders will be challenged.

2.2.6 Evaluation of the Email Anti-SPAM systems

In this section, a tentative for evaluating the above-mentioned Anti-spam techniques is provided. To achieve this task, criteria related to implementation, deployment, complexity, applicability to the SIP context and user convenience are used. The results of this analysis are described in

Technique	Advantages	Disadvantages
Prevention-based (server side) (technique used for spam prevention)	▪ Easy to implement ▪ Works for both calls and IMs ▪ No fake identity can be used for sending spam ▪ Can be very effective in spam mitigation ▪ Digital signatures which is a robust mechanism for enhancing security is a particular case of this technique. It can help in blacklisting and whitelisting	▪ It is not a filtering technique, however it can be used as a support for the filtering technique ▪ If digital signatures are used, they have to be implemented by all the providers
White lists (technique used for spam handling)	▪ Easy to implement ▪ Easy to install ▪ Works for both calls and IMs ▪ Can be implemented on a client or server, however, the first option is more convenient as the users have less difficulties to run filters on their computers ▪ Can be customized to the user preferences ▪ Medium filtering	▪ This technique is very restrictive if it is used alone ▪ Update is needed in order to allow new domains callers to reach the white list owner ▪ If this technique is poorly implemented, it might be bypassed by using spoofed IP addresses or fake domain names ▪ Medium to high rate of false positives
Black lists (technique used for spam handling)	▪ Easy to implement ▪ Easy to install ▪ Can be applied for both calls and IMs ▪ Can be implemented on a client or a server, however the second option is more convenient as the lists need to be updated frequently	▪ They can be bypassed by using spoofed IP addresses or fake domain names ▪ Anonymous SIP requests will be blocked ▪ Constant update is needed ▪ Medium to high rate of false positives

	▪ Medium filtering	
Content filtering (server side) (technique used for spam detection)	▪ Easy to install ▪ If it is well configured, this will allow a high rate filtering	▪ Can not be applied to phone calls ▪ They can be bypassed by modifying continuously the IM messages ▪ Constant update is needed ▪ The rules are static and do not change automatically ▪ Medium to high rate of false positives
Challenge/response (server side) (technique used for spam prevention)	▪ Can be applied for both calls and IMs ▪ Allows to distinguish between SIP requests issued by human being and machines ▪ High rate filtering	▪ Inconvenience for users if they are frequently challenged ▪ Can put users in trouble if they have to be challenged many times during a conference call ▪ It has to be used with other techniques such as white or black lists to be effective ▪ The challenges have to be updated frequently

Table 1 : Email Anti-spam techniques evaluation

3 Proposed spit techniques– state-of-the-art review

3.1 Proposed techniques

3.1.1 Spit Prevention using Anonymous Verifying Authorities

The authors in [2] present a new approach to prevent voice spam, by extending the call setup procedure. Their approach is based on the introduction of a "call-me-back" scheme, and the usage of two new entities in the IP infrastructure, namely: (a) the *Mediator,* and (b) the *Anonymous Verifying Authority*. *Figure 1*depicts an abstract representation of the proposed mechanism.

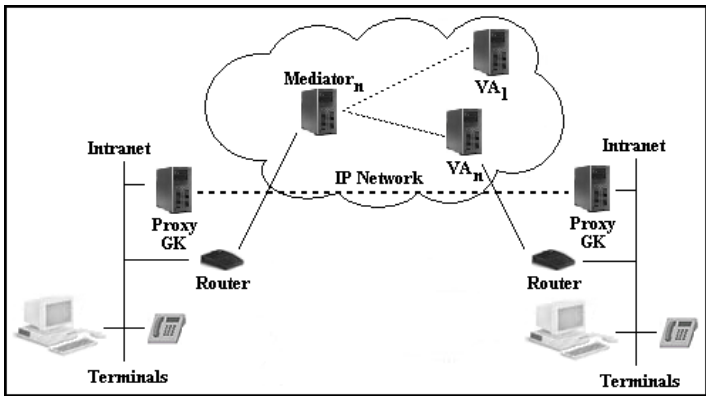

Figure 1: *Spit Prevention using Anonymous Verifying Authorities*

The role of the Mediator entity is to locate and forward call setup request information to a – Anonymous Verifying Authority (AVA), and to establish the requested session by joining the caller and the callee with a unique communication token. AVA is considered as anonymous to the caller. According to the proposed scheme, the Mediator receives call setup requests from the caller's Proxy server, and passes these requests to any Verifying Authority, available in the IP telephony infrastructure. On the other hand, Validating Authorities identify, validate, and locate the caller and the callee, request call-setup related policies, inform the callee proxy server, and generate a unique valid token in case the call was approved.

The scheme works properly under the assumption that the caller and callee proxy perform specific actions. More specifically, these proxies should only: send call-setup request to the Mediator entity, receive "call-me-back" and policy requests from a Verifying Authority, and, finally send and receive media streams. The latter action is performed if the caller has a pending call setup request and has a valid token assigned by the Authority.

To further describe the proposed mechanism, we present here a scenario in which two parties, Alice and Bob, registered on two different domains, () want to communicate.

1. Alice initiates and invites Bob to participate in a voice call.
2. Alice's Proxy server forwards this request to a Mediator which in turn forwards it to a randomly chosen Verifying Authority, keeping this choice secret from Alice.
3. The VA verifies the identity, validity, destination and location of both Alice and Bob. The VA may request any related policies regarding Bob's call setup requirements from Bob's Proxy server. During this period Alice will be prompted with a "ringing" tone.

4. At this point verification takes place according to Bob's profile, and if the session is approved, the VA sends a "call-me-back" request to Bob; otherwise the call is terminated. Three different outcomes are possible at this stage: (a) the call is terminated due to time expiration, (b) the call is accepted, or (c) rejects by Bob.
5. If Bob decides to accept the call, an "OK" response is sent to the VA. Then, VA generates a unique communication token.
6. The VA distributes this token to both Alice's and Bob's Proxy servers. Alice's Proxy will not be able to create the media session unless it maintains a call invite in a pending state, which is from Alice to Bob.
7. In this case the media session is established using the unique communication token.

3.1.2 Spit Mitigation through a Network Layer Anti-Spit Entity

The authors in [5] present a new approach to detect and mitigate spit through a network-level entity. The authors recognize that the adoption of anti-spit mechanisms either in the client-side or in proxy servers has many drawbacks. They propose the usage of the network entity in the edge of a network (i.e., access network points, WiFi access pointsetc). The basic function of the specific entity is to capture, filter and analyze the network traffic, passed from the equipment located in the edges of the network, in order to detect and mitigate spit calls. A generic preview of the proposed mechanism is depicted in *Figure 2*a. The functional components of the proposed anti-spit entity are illustrated in *Figure 2b*.

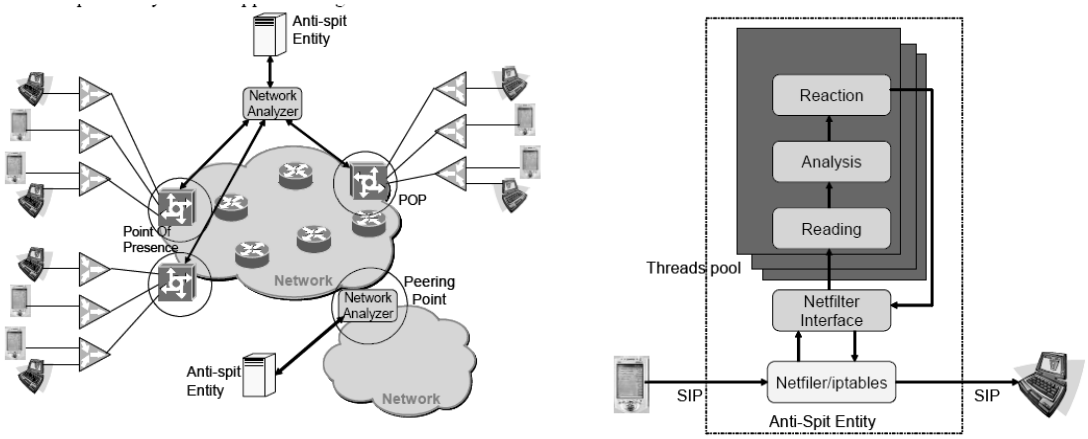

Figure 2: *(a) Generic Preview of Anti-Spit Mechanism, (b) Anti-Spit Mechanism Architecture*

The proposed mechanism activates filtering and analyzing at a first stage, regarding the traffic that receives from the network entity. This analysis takes into account only the SIP packets, while the rest of the network traffic is ignored. The goal of this procedure is to identify the users that participate in a SIP session, either by examine the users' SIP addresses (URI addresses) or the users' IP addresses (Care of Addresses). The next step is to classify if a SIP call is spit or not. The authors identified five criteria for detection, and they argue that this list is not exhaustive, since further criteria might be adopted. These tentative criteria are:

- The duration of the calls
- The number of received error messages
- The automated logic, examining if the destination address presented in the SIP headers follows a specific order-logic.

- The simultaneous calls attempts. A call is characterised as SIP if the number of the simultaneous calls made by the same user reaches a specific upper threshold.
- The call bombing analysis. A call is characterised as SIP if the number of consecutive calls made by a user reaches a specific upper threshold.

The classification of spit calls is given by the following formula:

$$spitLevel = \sum_{i} p_i \cdot a_i \text{ , for every i where } a_i = 1 \text{ when positive and } a_i = 0 \text{ when negative}$$

, where *i* is one of the abovementioned criteria and $p\backslash_i$ is the weight associated with each analysis a_i. If the *spitLevel* exceeds a specific threshold the call is classified as spit otherwise the call is further processed as normal.

The last task of the proposed mechanism is to perform handling actions in case a call is characterized as spit. These actions depend on the policies adopted by the domain where the specific user belongs, and the end-user's preferences. Some indicative actions include limiting the number of calls allowed, calls redirection, temporary blacklisting, etc.

3.1.3 Spit Detection based on Reputation and Charging techniques.

The work in [6] proposes two techniques to detect and mitigate spit: the first is based on a reputation scheme, and the second is based on a charging (payment) risk idea.

The reputation based technique has its origins in social networks and uses the amount of trust that the recipient (callee) has for the sender (caller) in order to distinguish if a message (call) is spit or not. The proposed approach is based on a reputation network. This is constructed by considering first the owner of a contact list and the users figured on this list as nodes connected with some scored edges, which express how much trust the owner of the list has to its members. *Figure 3* illustrates the overall architecture of the reputation-based techniques as well as its functional components.

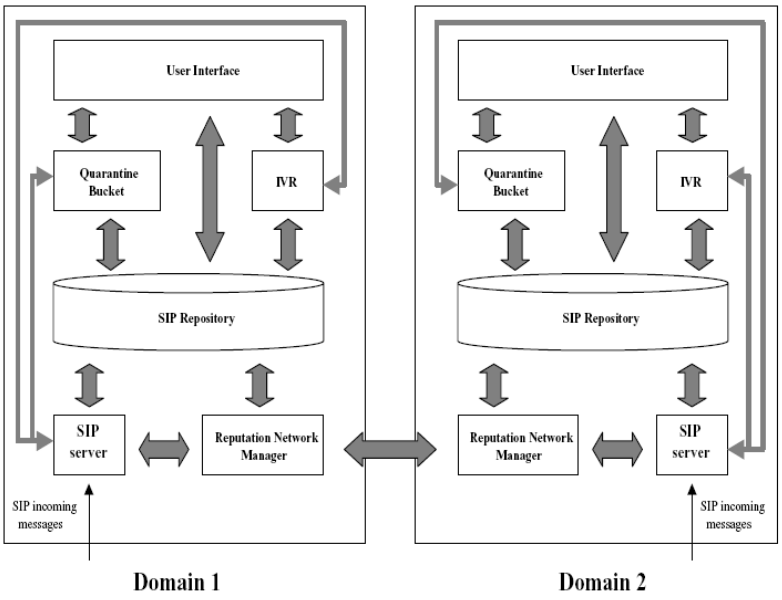

***Figure 3**: Reputation based technique architecture*

The main components of the proposed technique are:

- ***SIP Repository:*** This repository stores the contact lists of the different users along with the reputation values that each owner of the list has to its members. The database is distributed, allowing the exchange of information about the reputation values between different SIP domains (providers).
- ***SIP Server:*** This server is the main component of the proposed architecture. Its role is to intercept the SIP requests and forward the identities of the sender and receiver to the Reputation Network Manager (RNM).
- ***Reputation Network Manager:*** It finds out all possible paths in a reputation network between the receiver and the sender in a SIP message. In case the RNM cannot find a path in the receiver's domain, it will contact its neighbouring RNMs asking for entries in their repositories regarding the received SIP request's recipients. When the RNM finds a path it computes a reputation value corresponding to this path, and compares it to a predefined threshold. If the value is greater than the threshold the request is processed, otherwise is rejected.
- ***Quarantine bucket and IVR:*** The specific entities are involved if the SIP messages require further investigation to be classified as spit or not. When the RNM cannot assign a reputation value to a request, if that request is an instant message it will be forwarded to the quarantine bucket in order to be checked by the end-user. if the request is an SIP INVITE it will be forwarded to IVR and the media stream will be recorded in order to be checked by the receiver. In both cases, the receiver will check the message and will give a feedback to the system by assigning a reputation value to the sender.

The charging technique is based on the payment at risk solution where the senders pay for each message they send. Therefore, it is expected that the volume of spit will be reduced because the sending expenses will be increased, and might be higher than any possible benefit.

Additionally, the proposed technique assumes that: (a) the SIP users maintain a white list, (b) in each SIP server domain an AAA server manages users' authentication and charging actions (accounting services), and (c) some agreements are established between different SIP providers that facilitate the exchange of authentication information. *Figure 4* depicts the charging based mechanism.

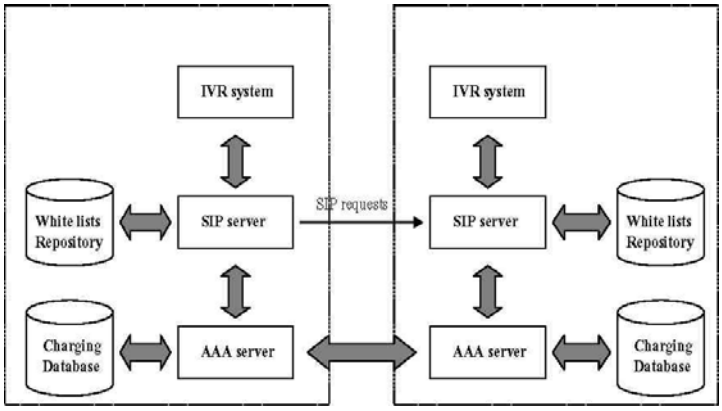

Figure 4: *Charging based technique architecture*

When the SIP server of the receiver's domain receives a SIP request, it first checks if the sender belongs to its white list. If so the SIP request is proceed. Otherwise, the SIP server

generates a reply, in a case the request was an instance message, or it forwards the request to the IVR in case the request was a session invitation (e.g., SIP INVITE). In both cases the goal is to inform the sender that he/she should pay a fee in order the request to be satisfied. If the sender refuses payment the SIP session terminates. If the sender finally pays the fee, then the SIP server passes the sender information to the AAA server which checks the authenticity of the sender and makes all the required charging and accounting actions.

3.1.4 DAPES

DAPES (Domain-based Authentication and Policy-Enforced for SIP) system [7] is one more system that tries to prevent spit. IT relies on the basic trapezoid SIP infrastructure to determine in real time if a call or an instant message is likely to be spit or not. DAPES is based on domain descriptions and reputation systems

It assumes that the following requirements are fulfilled:

- All the SIP messages are passed through the proxies of the domains that the caller and callee belong, and implementation of TLS or IPsec authentication is mandatory.
- Outbound proxies must have certificates signed by well-known CAs.
- Calls are routed through outbound proxies belonging to the caller's domain.
- All signalling association between proxies must use TLS or IPsec.
- The identity of the caller must be verified by its domain proxy.

As we mentioned above DAPES performs domain-based authentication and verification of incoming calls, and is assisted by a reputation system to determine if calls are spit. Domains are classified based on their trustworthiness, and the probability of affiliated users sending out spit calls. The criteria used to classify domains are associated with identity management, authentication, authorization and accounting procedures. The recognized categories are: (a) admission-controlled domains, (b) bonded domains, (c) membership domains, (d) rate-limited domains, and (e) open domains.

In DAPES any SIP-based communication enter in two stages of verification. The first stage deals with the verification of the caller's identity, which could be done in two different ways: (a) with digest authentication of the SIP INVITE message, and (b) with digest authentication of SIP REGISTER and SIP INVITE message, and address verification of the SIP INVITE message. The second stage of verification involves the mutual authentication of the participated proxies through TLS, and the verification of the outbound proxy through the DNS Service Records. Furthermore, in DAPES some usage scenarios have been identified, based on the domains classification, in order the known users to go through simple authentication and verification procedures, whilst unknown users to deal with stricter procedures. Those scenarios are:

- ***Known Users***: Users that are known to the caller through personal relationships or previous communication (e.g., incoming call logs) are placed on whitelists. For calls made by these users the only requirement for further processing is the authentication of the user.
- ***Unknown user, trusted domain:*** In case the caller has been authenticated by its domain's outbound proxy but he/she is not known to the callee, the outbound proxy should request from a peer proxy or from an independent SIP provider to assert the caller's trustworthiness..

- ***Roaming Users:*** In this case the user should route its messages through its home proxy, and authentication is performed as described in the first case.
- ***Unknown user, unknown domain:*** In this case it is important to retrieve information about the trustworthiness of the users and their domain. For that reason a SIP-oriented reputation system is defined. This reputation system use the concept of "karma" used in community web sites or seller/buyer ratings on auction sites as eBay while is implemented as a set of distributed servers that keep lists of all registered users, domain and their reputation ratings. The message flow in this case, in depicted *Figure 5*.

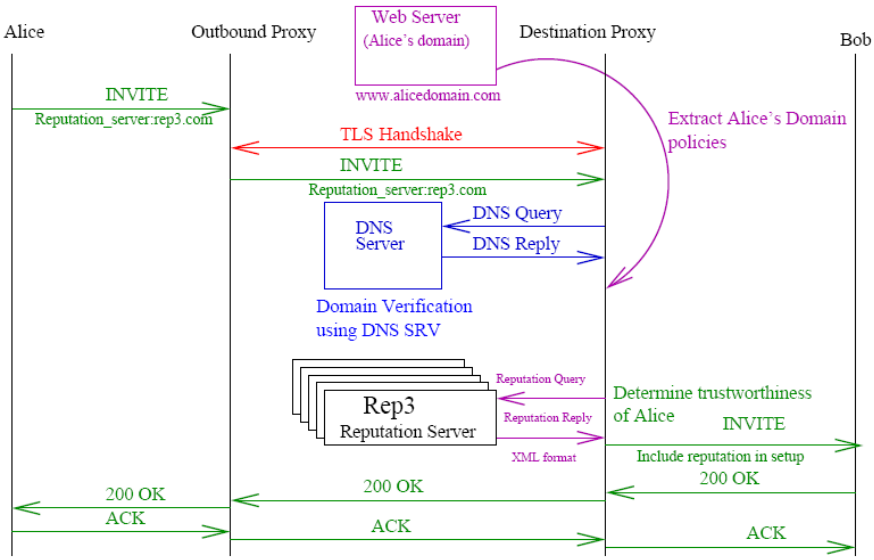

Figure 5: *Usage of the DAPES' reputation system*

3.1.5 Progressive Multi Gray-Leveling

A novice spit protection algorithm called *Progressive Multi Gray-Leveling (PMG)* has been proposed in [3]. Basically, the algorithm calculates and assigns a *gray level* for each caller, which determines whether the caller is a likely spam source or not, and based on that level decides if a call is to be established. The level is calculated based on previous call patterns of the particular caller, rather than the feedback from other users.

The notion of a gray level stems from a relevant spam email protection algorithm, where the term *gray* indicates a level existing between white and black. Whereas the blacklisting technique blocks all mails from a given source and the whitelisting accepts all from the source, graylisting determines the legitimacy of a sender depending on current information. However, different from graylisting that investigates the content of an e-mail, PMG monitors the call patterns from each caller and identifies spit based on these.

The idea of the PMG algorithm is that as a caller attempts to initiate numerous calls through the call server in a certain time span, his/her gray level will increase, thus classifying him/her as a potential spam source. Once the gray level becomes higher than the given spit source threshold, the caller will not be able to initiate any more calls. However, the graylist differentiates from a typical black list in that the caller will not permanently stay in a "spit status." Thus, if the spit caller stops initiating calls within a certain time period, the gray level

will gradually decrease and eventually drop below the given threshold and the caller will be allowed to initiate calls.

In application, PMG calculates not one, but two levels: a) a short-term, and b) a long-term one, both based on the caller's call patterns. PMG identifies a call to be spit, when the summation of the two levels exceeds a given threshold. As long as this summation appears less than the threshold, the caller is flagged as a regular user and, thus, his/her call is established. If the summation gets greater than the threshold, the caller is flagged as spit generator and his/her calls are therefore blocked.

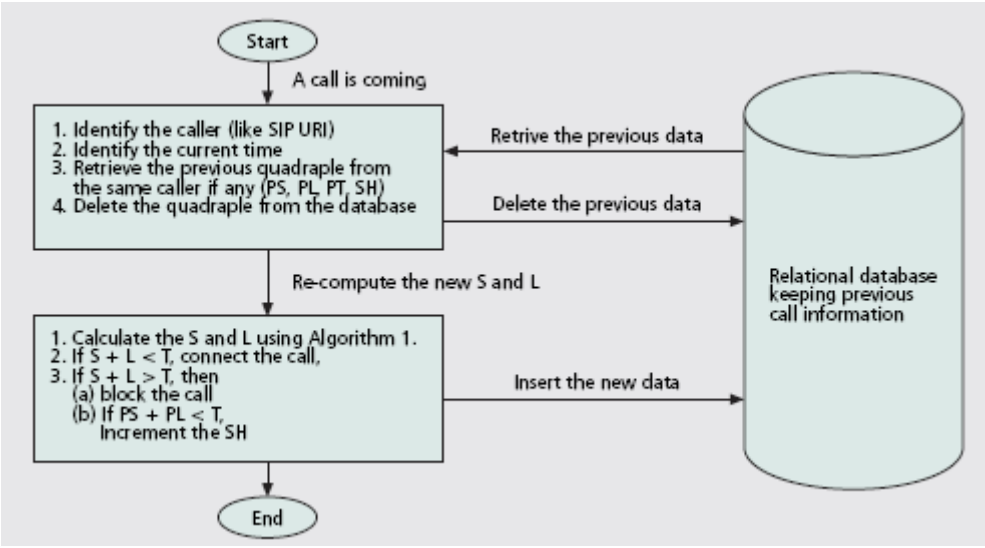

Figure 6: Progressive Multi-level Graylisting

[3] implemented PMG on three platforms: a) the Cisco Call Manager using the Java Telephone application programming interface (JTAPI), b) the Vovida Open Communication Application Library (VOCAL) server using stateful proxy and c) IPTEL SIP Express Router (SER) using stateful proxy. In the experimental application of PGM, the short-term level represents a short period of time (e.g., 1 min) during which a spit source is able to generate many calls to attack a server and it reflects the current behaviour of the caller at that small time frame. The short level increases very quickly in this case, so as to protect the server by blocking quickly these intensive, but normal, calls. The level decreases as soon as the caller stops initiating calls. If the caller does not make a call in a couple of hours, the short-term level returns back to zero. Hence, using only a short-term level, a spitter is able to generate spit over again after a relatively short period of time.

To compensate for the short-term level issue, they use in combination a long-term level that reflects the call patterns over a rather long period of time (e.g., several hours or days). The long-term level increases more slowly than the short-term level and also decreases at a much slower pace. At the same time, it also takes into account the history of a caller that has been detected as a spam generator. If a caller has ever been detected as a spit source, the long-term value is multiplied by the number of times it was detected as a spitter and increases much faster than other regular users. Hence, a spam generator is able to make only a fraction of the calls it originally produced in its previous trial.

PMG's key element is to define the proper thresholds and its main drawback is that it identifies a caller by its ephemeral address, such as a SIP URI. So if a spitter can alter its

address easily, PMG does not provide concrete solution, since it has to restart the gray level computation over again.

3.1.6 Biometric Framework for Spit Prevention

As already identified one major aspect of the spit problem (as well as the e-mail spam problem) is that spammers can change their identity frequently. To address this, [1] propose the use of global servers that bind the users' identities to personal data; they select biometric data, such as a person's voice. The servers are selected as global, so even if a caller switches to another SIP provider, he may be still identified.

In their rather generic framework, they use a set of trusted authentication servers (A), with which a client (C) has to register, when he first uses VoIP. The goal is to record the user's voice and bind it to his/her VoIP ID. First, the client calls the server. The server then asks the client to repeat a sentence, for example, a phrase from Goethe's Faust. In order to enhance the security of this procedure, the phrases are different for each registration request. Moreover, several different languages are offered, such that each client can use his mother tongue. Then, the server stores the client's voice file and sends back the credentials; in case of a PKI infrastructure, this is a server signed public key (Step 2). The client can now make arbitrary calls to other clients, authenticating himself using his credentials (Step 3). Anyone receiving such a call verifies the identity by checking the credentials; this may involve contacting the authentication servers (Step 4).

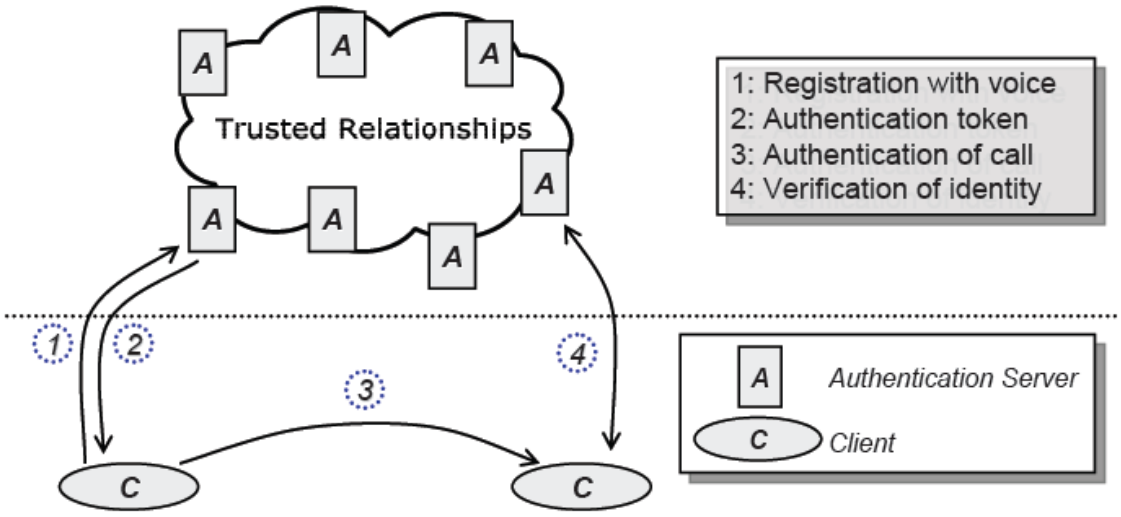

Figure 7: Biometric framework for spit prevention

The key idea here is that it is impossible for a client to run a Sybil attack: A client who wishes to obtain additional identities is unmasked by the authentication server: The servers run a voice recognition software to reject duplicated registrations. The approach is interesting as a) it is independent of a specific VoIP protocol and inter-operable, and b) it is independent of an ISP, as an attacker cannot obtain new identities by switching to another provider. One as to note that the biometric data are not used for the authentication of a user, but only as reference data for future registration requests. To implement this idea, [1] describe an implementation using a public key infrastructure (PKI) and an implementation using Kerberos.

However, the initial authentication procedure and the process of distinguishing voices of users can be proven time consuming or difficult, as voice patterns are sometimes close to each

other, and patterns can also be changed, for instance by using some background noise. Another issue is that the certification servers must be powerful in order to avoid bottlenecks and the use of personal data has always several problems, considering security of the stored data and the community acceptance.

3.1.7 RFC4474

The main objective of this RFC is to address the end-user authentication issue, when they initiate SIP requests, especially in an inter-domain context [13]. The proposed authentication scheme aims to provide end-to-end authentication (i.e., UA-UA), whilst the identity control is achieved within the domain that serves the initiator. This RFC tries to enhance the security level, in terms of authentication, provided by other mechanisms including Digest, TLS, and S/MIME. Strong authentication, when provided, ensures the called party about the identity of the calling party, and, thus, spit traffic could be blocked at the receiver.

The RFC uses two new SIP header fields:

- *Identity*, for conveying a signature used for validating the identity, and
- *Identity-Info*, for conveying a reference to the certificate of the signer

Only a proxy Server manipulates the *Identity* and the *Identity-Info* fields. This is achieved within the domain of the calling UA through the usage of digest authentication, and domain certificates. Domain certificates are used as a mean to overcome scaling objections introduced by the end-user certificates.

Imagine a scenario where Alice wishes to communicate with Bob and they are both located on different domains.

Alice

1. She generates a SIP *INVITE* message and places her identity in the *From* header field.
2. She sends the SIP *INVITE* over TLS to an authentication service proxy of her domain.

Alice's proxy

1. The proxy Server includes an authentication service that authenticates Alice (e.g., by sending a Digest authentication challenge) and validates that she is authorized to assert the identity that is populated in the *From* field.
    a. This value may be Alice's AoR, or it may be some other value that the domain policy permits her to use.
2. It then computes a hash over some particular headers, including the *From* header field and the bodies in the message.
    b. The fields covered are *Contact*, *Date*, *Call-ID*, *CSeq*, *To*, and *From*
    c. This hash is signed with the certificate for the domain and inserted as a new header field in the SIP message, the *Identity* header.
    d. The proxy, as the holder of the private key of its domain, is asserting that the originator of this request has been authenticated and that she is authorized to claim the identity (the SIP AoR) that appears in the From header field.

e. The authentication service of the proxy usually ensures that any *Date* header field in the request is accurate, as well

3. The proxy also inserts the *Identity-Info* header field, that tells Bob (or his domain) how to acquire its certificate, if he doesn't already have it. For instance this field might contain a URI from which proxy certificate can be acquired.
4. Forwards the SIP *INVITE* request that includes the added header fields

Bob's proxy or UAS

5. When Bob's domain receives the SIP *INVITE* request, it verifies the signature provided in the *Identity* header, and thus can validate that the domain indicated by the host portion of the AoR in the *From* header field authenticated the user, and permitted the user to assert that *From* header field value.

The above specification applies to other methods, such as SIP BYE and SIP REGISTER, as well. Additionally, RFC4474 proposes three new SIP response codes:

- When the verifier proxy receives an *Identity-Info* field containing a URI that cannot be dereferenced (either the URI scheme is unsupported by the verifier, or the resource designated by the URI is unavailable) a *'Bad Identity-Info'* response is initiated backwards.
- When the verifier cannot validate the certificate referenced by the URI of the *Identity-Info* header, because, for example, the certificate is self-signed, or signed by a root certificate authority for which the verifier does not possess a root certificate a *'Unsupported Certificate'* response is generated.
- When the verifier receives a message with an Identity signature that does not correspond to the digest-string calculated by the verifier a 'Invalid Identity Header' response code is initiated.

The RFC4474uses some concepts originally provided in the S/MIME-SIP implementation (RFC3261, Section 23). It does not protect against man-in-the-middle-attacks, since the *Identity*, and *Identity-Info* fields are not protected (signed). It does not seems to offer many more guarantees and security services than an end-to-end TLS with client/server authentication. Scalability is a question, since domain certificates do not actually remove scale obstacles, whilst cross-certifications and long certification paths are essential. Certificate management is also an issue, since real-time Certificate Status Information is required. Additionally , verification process might delay the session initiation phase.

3.1.8 SIP SAML

The Security Assertion Markup Language (SAML) is used for the expression of security assertions, such as authentication, role membership, or permissions. For example, SAML assertions have been used to realize single-sign-on between Web servers located in different domains. A SAML assertion encodes security information about an entity and may contain assertion statements, like authentication statements (e.g., when, by whom, via which authentication mechanism), attribute statements (e.g., the department in which the subject works), or authorization decision statements (e.g. the subject has permission to access a particular resource).

The use of SAML and form a generic solution for SIP authentication and authorization is proposed in [8]. In this work a logical role of a SIP Authentication Service is introduced, typically offered by a SIP outbound proxy. SIP user agents send requests through an Authentication Service, which:

a. Authenticates the user according to a set of practices

b. Creates and cryptographically signs an authentication token for the user

c. Shares that identity to others

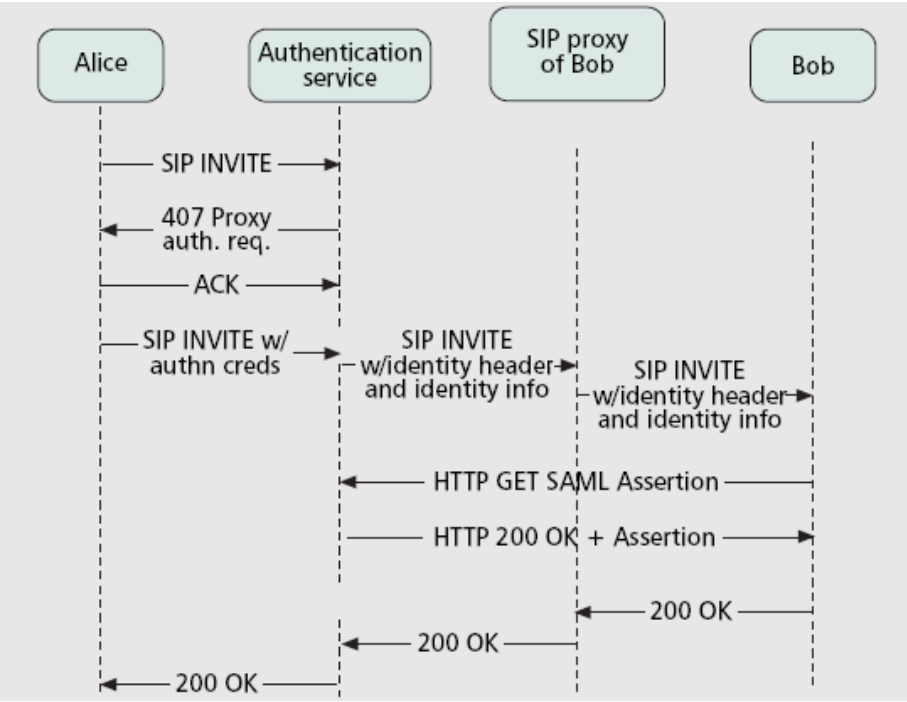

Figure 8: *SIP SAML usage*

Figure 8 illustrates an example in which Alice initiates a call to Bob. Alice authenticates with the Authentication Service, which forwards her SIP INVITE message on to Bob's inbound SIP proxy. This SIP message includes Alice's identity information, provided by her proxy, a reference to a SAML assertion, which asserts various traits of Alice, and points to Alice's domain certificate. If the assertion and domain certificate pass Bob's verification process preformed in his inbound proxy, then the call setup continues. More detailed:

Steps 1 and 2: an authentication and authorization process is executed between Alice's SIP UA and the Authentication Service. This is important for the creation of the SAML assertion and the respective attributes. The SIP UA must ensure that the Authentication Service is genuine.

Step 3: the Authentication Service verifies the identity information in the SIP INVITE message of Alice's UA and generates a SAML assertion attesting to Alice's various attributes, according to a local configuration. The assertion is held to be retrieved later retrieval by the Relying Party, and the SIP INVITE message is modified. This includes placing an HTTP URL reference to the aforementioned SAML assertion in the "identity-info" header field of the SIP INVITE message.

Step 4: Bob receives the SIP INVITE message and it extracts the URL from the identity-info field and dereferences it using an HTTP GET.

Step 5: the SAML assertion is returned in the HTTP response message, Bob receives it and verifies it, and the domain certifies the assertion references.

Step 6: if the verification of step 5 succeeds, a SIP response with a success status code ("200 OK") is returned to Alice's UA, and call setup proceeds.

A SIP message, such as an INVITE, may traverse zero or more intermediaries, any of which may be untrustworthy. Since a SIP INVITE message actually contains an HTTP URL with the associated SAML assertion, any of the participating entities are able to retrieve the assertion and the associated domain certificate. Additionally, since the HTTP-based infrastructure usually involves proxies, one of those entities could intercept a returned assertion. The attacker could then conceivably attempt to impersonate the subject (e.g., Alice) to some SIP-based target entity. However, the attacker would not have the corresponding private key with which to generate the signed SIP Identity header value. Also, due to the assertion content, termed as a "SAML assertion profile," the assertion will not be useful to arbitrary parties, because:

• Is digitally signed, thus causing any alterations to break its integrity, making them detectable

• Does not contain an authentication statement

• Identifies the targeted relying party

• Identifies the assertion issuer

• Explicitly stipulates its validity period

• Contains or refers to the originating user's domain's public key certificate

3.1.9 DSIP

DSIP screens spit by classifying incoming callers, where each classification of a caller applies to different sets of privileges. In DSIP, a callee can classify callers into one of three classes: a) whitelist (regular contacts), b) blacklist (known VoIP spammers), and c) greylist (neither regular contacts nor known spammers) and handle the corresponding communication differently [4].

• Whitelisted callers benefit from no restrictions and all privileges are available, with respect to contacting callees and leaving voicemail messages. These callers are assured that their call flow will not be interrupted by DSIP, and these callers also have the ability to leave voicemail messages directly on callee's machines.

• Blacklisted callers have no ability to contacting callees or leaving voicemail messages. These callers are assured that their call flow will be interrupted by DSIP and they will be unable to communicate.

• Greylisted callers have to first pass a human verification test. Greylisted callers are directly rejected if they cannot pass the human verification test. After passing the test, they can operate in a restricted environment. They can communicate with the callee, however, they cannot leave voicemail messages on the callee's voicemail server, if the callee is not available for the calls. Instead, they must store the actual message on their own machines, and provide the callee with instructions such that a callee can retrieve the message later at their convenience, in the same way how unclassified senders send email in DTMP (Differentiated Mail Transfer Protocol) proposed in [9].

.

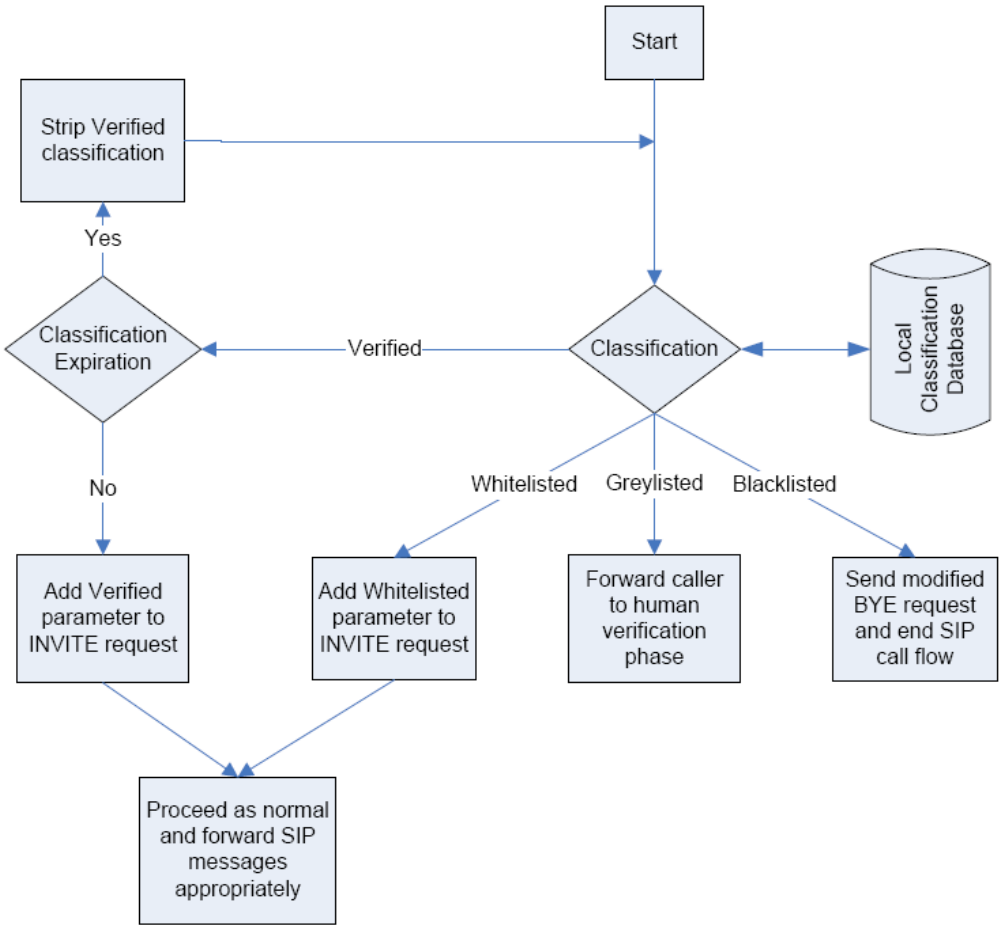

Figure 9: Differentiated SIP

DSIP provides end users with flexibility over who can contact them, and assist on mitigating voicemail denial of service attacks.

3.1.10 Voice Spam Detector

The Voice Spam Detector (VSD) system [12] constitutes a framework based on many anti-spit techniques in order to handle spit phenomenon in a more efficient way. More specifically, the system is a multi-stage spit filter based on trust and reputation, while using closed loop feedback between the different stages. The main building blocks of the VSD system are depicted in Figure 10. The next paragraphs we present a brief description of these blocks.

- Presence Filtering: Presence filtering depends on the current status of the callee. For example if a callee is in a “busy” status then she might not accept an incoming call or message. Hence, the first step of the VSD is the characterization of spit in accordance with the callees’ state that is based on specific predefined rules, similar to firewall rules.

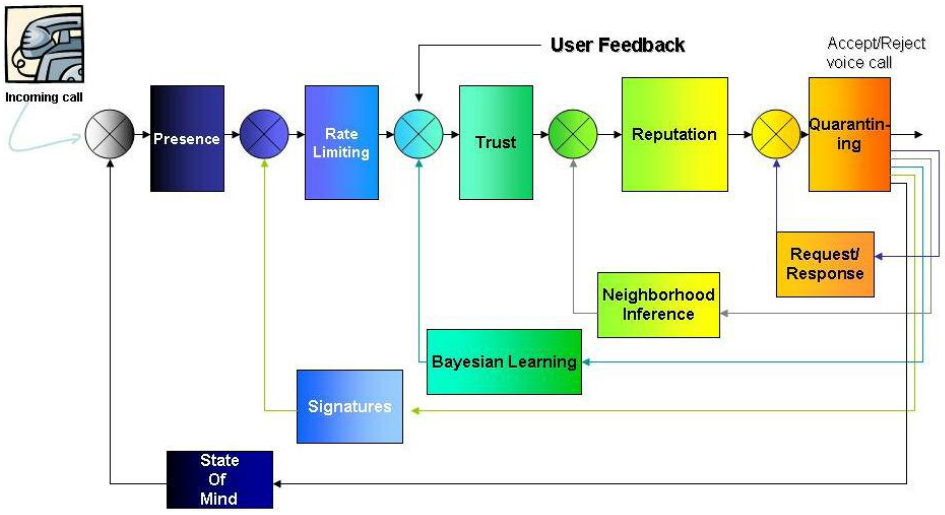

Figure 10: VSD System building blocks

- Rate Limiting: In this stage incoming calls are checked based on traffic pattern analysis. More specifically, the rate of the incoming calls initialized by a specific caller or a sIP domain is checked against allowed incoming rate thresholds. If the incoming call volume is higher than the predefined threshold, then the call is blocked, otherwise is permitted.
- White/Black Lists: In this stage the well-known white/black listing filtering is used. Therefore, if a caller belongs to the white list of the callee then the call is accepted; otherwise is blocked.
- Bayesian Learning: In this stage a call is checked regarding the behavior of the participated entities. More specifically, the behavior of the participating entities is estimated during a period of time, and future trends can be extracted by historical data of call attempts to the end-users of the called party's domain. VSD checks for potential spit behavior associated with any of the participating entities, by looking up trust information that is available for those entities. The spit probability of a call C, P(X|C=spam), can be computed using Bayesian inference techniques. In the context of VSD the following formula is used to compute the spit probability of an incoming call:

$$\frac{P(C = spam) \prod_{i=1..n} P(x_i = spam)}{\sum_{k=spam,valid} P(C = k) \prod_{i=1..n} P(x_i = k)}$$

  where each of $x_1,...,x_n$ represent different header fields of a SIP message, (e.g., "From", "To", "Via" "Record Route", and "Contact Info"). VSD filters out a call if its spit probability of is greater than a tolerance level. Otherwise, the call is forwarded to the actual recipient. In that case, the VSD waits for a feedback from the recipient. The tolerance level is chosen by giving a preference of valid calls over spit calls i.e., the number of spit messages that are admissible so as to minimize the blocking of valid calls (False Positives).
- Social Networks and Reputation: The final stage of VSD is based on the social networks that can be used to represent user relationships. These could be derived along the paths of the network. The user's social network represents the associated and trusted neighbors from which the user is willing to receive calls.

3.1.11 VoIP SEAL

NEC Corporation announced the prototype of VoIP SEAL, which defends against spit.. Its main feature is the adoption of a module structure, an idea that enables rapid response to new kinds of spit attacks, without adjusting the system, by adding and updating modules that respond to new and different kinds of spit. The concept, presented by Niccolini in [10], uses a two-stage system: the first stage has invisible (non-intrusive) modules and the second supports interacting (intrusive) ones. Each first stage module attributes a score in [-1,1], where high score represent high probability that the call is spit. Each module is associated with a weight, and the total score is compared to two thresholds. If the score is higher than the lower threshold, then the call passes to the second stage modules, which could be a Turing test and if the score is higher than the higher threshold, the call is rejected. Feedback mechanism is used on a per-user basis to blacklist users.

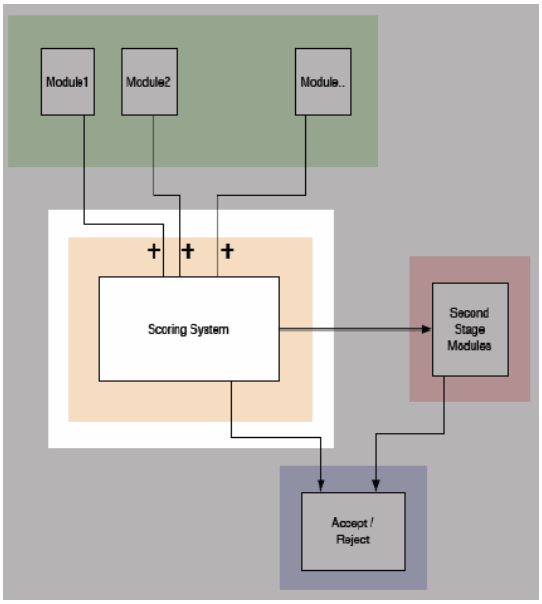

Figure 11: NEC module system

The adoption of a module structure also realizes response to a broad range of applications by enabling flexible and easy customization of systems to meet the needs of a variety of hardware, such as SIP servers, Session Border Controllers (B2BUA in SIP), home network equipment and terminal equipment. Some concepts implemented in Voip Seal can be also found in [11]. The authors propose an IDS/IPS architecture specific for VoIP applications. They suggest using network-based intrusion detection and prevention systems with a two-stage technique (first apply knowledge-based techniques and then apply behavior-based techniques on the packets passing the knowledge-based step in order to reduce false positives). A prototype implementation of the proposed architecture was realized based on Snort software. This prototype implements SIP knowledge-based security checks and poses the basis for the implementation of behavior-based ones thanks to a stateful analysis.

4 Definition of efficiency criteria for spit avoidance

4.1 Attacks scenarios exploiting threats and vulnerabilities

Several threats regarding spit have been identified, that could lead an attacker to the opportunity of delivering a spit violation. These threats were analyzed in the previous deliverable (section 5.1), and categorized using as a point of reference their relationship to the specification of the SIP protocol.

In order for the reader to understand how the attacks are likely to be delivered and how she/he will exploit the SPAM oriented vulnerabilities of SIP, four (4) general categories of attack scenarios were identified and are presented in this section. The scenarios have been categorized according to the time frame of the execution of an attack:

Step 1: Identifying of spit targets,

Step 2: Examining possibilities of bulk sending of messages and maximization of profit

Step 3: Identity hiding

Step 4: Spit message sending and delivery (PITM attacks, spit messages in SIP headers and bodies)

These categories contain a subset of the previously identified threats, since this section's scope is to provide a theoretical demonstration of the actual implementation of spit attacks in SIP environments, and not an exhaustive list of all the possible permutations of the spit oriented threats. According to these timed categories, the attacker first collects target addresses. Then she/he attempts to exploit SIP vulnerabilities that could permit the sending of bulk messages without redundant resources, while hiding her/his identity. Finally, the attacker aims at the actualization of another subset of spit threats, with an eye towards sending the actual spit messages.

It must be noted that these scenarios are not exclusive, but rather complementary, since several of these attacks can be issued sequentially for the execution of a specific spit violation attempt. Additionally any of the following categories and specific scenarios can be permutated in any numbers of steps for a complete list of all possible attack scenarios deriving from the abovementioned spit threats.

4.1.1 Identification of targets

First, the attacker will probably try to collect addresses of possible targets for sending the spit messages. SIP specification provides several alternatives and makes the process of gathering victims' addresses on a SIP network rather trivial. The attacker has just to complete one or more of the following steps in order to populate the attack list.

- In a network where hosts use the SIP multicast address "sip.mcast.net" (224.0.1.75 for IPv4) for sending Registrations, the spammer listens to that address for a particular period of time, collecting information about the locations of other users. The longer the period, the bigger the attack list that she/he populates.

- The attacker sends an SIP INVITE message to a proxy server in the network, with the string "a@domain.com" in the Request-URI field. The proxy has no record of a user registered with that particular SIP URI and returns a 485 Ambiguous response, in which the Contact header field lists a number of users' addresses whose names start or contain the character "a". The spammer, depending on the number of addresses returned, may repeat the attack with a different string this time for concentrating more users' contact information.
- The spammer is running her/his network card in promiscuous mode, capturing SIP, or even IP packets on a communication path near a Registrar or a Proxy. Next, she/he processes the received packets, extracting the To, Via and Contact header fields from the SIP messages, creating concurrently a list of possible spit messages recipients.
- The spammer uses a port scanning tool (e.g. nmap) targeting only the ports 5060 and 5061. These well-known ports used by User Agents for SIP communications, ease the detection of possible spit targets.

4.1.2 Examination of possibilities of bulk sending of messages and maximization of profit

Spammers try to maximize their profit, either by a) sending large amounts of messages without increasing linearly or exponential their resources' cost, or b) by increasing the likelihood of message delivery. In the SIP context, the attacker may succeed in both the above ways, as shown in the following scenarios:

- The attacker sends the spit message in a multicast address so as to target multiple victims with only one message. The only action that she/he must take is to identify one such address and encapsulate it in the Via header field of the spit message identifying it with the "maddr" parameter.
- In a network topology, one or more forking proxies might be present in order to increase the quality of services provided to users. This increment is fulfilled by minimizing the time of deliverance and the amount of messages exchanged during the identification process of the route path to the message recipient. More precisely, if an ambiguous address is provided to the forking proxy, the proxy takes the initiative and forwards the message to all the possible recipients returned by the location service database search engine. After identifying a forking proxy, the attacker just sends the spit message to it, putting as recipient URI the string "a@domain.com". The location service returns to the proxy a large list of possible users, and the proxy in turn executes the attack for the spammer, by sending the message to all the recipients mentioned in the list.
- The spammer identifies a spit detection mechanism that examines all the SIP messages exchanged in a network. Moreover, many users are using a similar mechanism that checks all the messages received by the user agent. This examination is real-time and for that reason, the particular mechanism forwards urgent messages (mentioned in the Priority header field of the message) to the user without a check, since even the smallest delay could have a great negative impact for the user. The spammer exploits this vulnerability, by sending her/his spit messages with high

priority header fields so as to bypass as many spit detection mechanisms as possible, and deliver the message to the recipient with a higher probability.

4.1.3 Possibilities of identity hiding

Spammers will most certainly try to protect themselves and their identities in a twofold way. By hiding their tracks and by directing the forensics efforts towards other systems rather than their own. In a SIP environment the attacker will try to abuse vulnerabilities as described in the following scenarios:

- In the Email paradigm spammers relied on Open Relay Servers to forward unquestionably their messages to the intended recipients. SIP spammers will use stateless proxies in a similar way, since their messages and requests will be forwarded, without authentication. Thus, an attacker after identifying a series of stateless proxies will form a path route that chains two or more of them. Since the proxies will virtually hold minimal information for the messages exchanged through them, the spammer will increase the possibilities of track hiding.
- Stateless proxies are not the only SIP entities that an attacker will try to abuse. SIP Back-To-Back User Agents (B2BUAs) are ideal candidates for implementing anonymizing services. In a network providing such services the spammer will try to send the spit messages through them, so as to cover her/his tracks and true identity. Further combining stateless proxies and B2BUA anonymisers in the route path of the messages will make the identification process more resource demanding and difficult.
- The attackers will use either bots or some of their resources for instantiating several SIP accounts in one or more domains. This way, spammers add another level of difficulty not only for their identities detection process, but also for the detection of the actual spit messages since mechanisms and techniques for detection (e.g. Black or White Lists) become obsolete.

4.1.4 Spit message sending and delivery

A spammer has various weapons in his arsenal for delivering a successful spit attack. Among them, means that actually deliver the spit message to the intended recipients. The spit message is sent to the end user, by being routed through a series of elements, including the sender (the one end of the communication chain), several proxies (the chain) and the recipient (the other end of the communication chain). This chain is as strong as its weakest link. In the SIP example, this means that the whole chain can be compromised by injecting a spit message in any of the abovementioned links (sender, proxies and recipient). Since importing the spit message by compromising the recipient's agent is extremely cost demanding for a large amount of messages, the attacker aims at the other two links. More specifically, he will use means to either send the spit message, or take over a proxy to deliver the attack.

As far as proxies are concerned, they can be mitigated rather easily since a cornucopia of vulnerabilities is available freely on the internet, in several vulnerabilities databases which are enriched daily. Thus, after taking over a proxy, the injection of spit messages in the network is rather trivial. However, as shown in the previous threats analysis section, SIP specification provides the attacker with additional attack possibilities. These attacks are described below:

- The attacker has managed to take over a proxy. She/he now records all communication pairs of users' addresses that send their requests through this particular proxy. The attacker waits for each pair of parties, some short predefined time, which she/he has calculated as the threshold for the beginning of the media exchange between the two ends of the communication. Since the SIP messages have been exchanged, and the media exchange has start, the communicating parties are confident that no spit message can be sent. However, the attacker sends two SIP Re-INVITE messages, one for each user, spoofing the originator's address of the message with the address of the other user. This message modifies the session by adding one media stream that contains the spit message. As a result, both users receive the spit message, without knowing who send it to them.
- In the above scenario, the attacker instead of waiting for the media communication to begin, she/he spoofs all the SIP requests exchanged, by adding a Record-Route header field that contains the proxy's address, thus managing to remain in the SIP messaging path beyond the initial SIP INVITE. The attacker, through the proxy, is now in all the communication paths established through her/him. Next, the spammer tries to implement some mid-calls features that speculates that could be supported by the recipients. In the cases that this holds true, the SPIT message is delivered successfully.

If the attacker decides that the proxy mitigation attack is not cost effective for their intentions, they will send the spit messages using their own means. Spamming can take place even if the media session is not established between two communicating parties. It can be achieved through SIP messages' bodies or header fields. Several such possibilities have been identified and presented in the threats analysis section. Next are presented two indicative example scenarios.

- The attacker constructs a large amount of SIP messages that contain in the From header field the spit text message. Then, he spoofs the return address so as not to be easily identified from the recipient and forwards them to all the members of the attack list. When the messages are delivered to the targets, the From header fields are rendered by the User Agents and thus the spit message is displayed to the user.
- The attacker constructs a large amount of SIP messages, only this time he encapsulates in the messages' bodies the spit messages in the form of a picture, so as to avoid detection from the anti-spamming mechanisms that could be used by the recipients. Next the attacker inserts a Content-Disposition header field with the value "icon" and sends the messages to all the possible recipients. When the SIP messages arrive to their targets, the User Agents interprete the message, and display the spit picture to the end user, as denoted by the corresponding header field.

4.1.5 Examples of multi-step spit attack scenarios

The following hypothetical scenarios describe the steps that the spammer could take in order to issue a spit attack. In the first scenario, it is described how an attacker may attempt to harvest the addresses of the victims to the actual spit attack violation.

The attacker sends an SIP INVITE message to several proxy servers in the network, with the string "*@proxy_domain.com" in the Request-URI field. Some of the proxies are purely

configured and return a 485 Ambiguous response, in which the Contact header field lists several users' addresses.

Concurrently the attacker scans the network for forking proxies. The scanning results identify some forking proxies in the same domains with the previously misconfigured servers. Thus, until now the attacker has collected a rather large list of users in several domains, and a list of domains with forking proxies and misconfigured proxy servers (regarding the search string "*@proxy_domain.com").

Next the attacker constructs a large amount of SIP messages with the following properties. The return address is spoofed so as both to avoid detection and not to receive any replies that could block her/his system. Additionally she/he encapsulates in the messages' bodies the spit messages in the form of a picture, so as to avoid detection from the anti-spamming mechanisms that could be used by the recipients. Next the attacker inserts a Content-Disposition header field with the value "icon".

Finally the spoofed spit messages are sent to the users' list that was earlier populated, and to the forking proxies list. The messages sent to the forking proxies contain in the To header field the string "*@proxy_domain.com", so as to automate the message forwarding to all the users that the proxy has in its records.

When the SIP messages arrive to their targets, the User Agents interpreter the message, and display the SPIT picture to the end user, as denoted by the corresponding header field. No additional information is revealed about the attacker's identity, and thus a successful SPIT attack has taken place.

Another alternative hypothetical scenario that mainly relies on the hijacking of a proxy server by the spammer is the one described below. The scenario is ideal for covering attacker's tracks.

The spammer after counselling one of the many vulnerabilities databases on the internet discovers the exploit that will enable her/him to finally take over one of the main proxy servers in her/his domain. After gaining control of the proxy, she/he starts immediately to log, all communication pairs of users that send their requests through this particular proxy.

The spammer has already constructed a template of a SIP INVITE message, in which the From and To header fields are blank, the Subject header field contains the spit text message, and the Priority header field is set to important. The reasoning behind the Priority header field is twofold. Firstly the attacker has greater possibilities of bypassing a possible detection mechanism set at the Users Agents (as explained previously), and secondly the header field Subject has greater possibilities of being rendered and displayed to the end-users, since the User Agent's software might be tricked to ask the user for an appropriate action to this "important" received message.

For each pair of users that establish communication (the attacker can identify it from the messages being exchanged, through the automation of the process of log watching), the attacker constructs two SIP Re-INVITE messages using the abovementioned template. In the first message she/he fills, in the From header field the first user's address, and in the To header field the second user's address. In the second message she/he puts the addresses in a reverse manner to the corresponding header fields. Finally the attacker forwards the two messages to their destinations.

After the messages are received by the end users, the User Agents render the subject to the each user as their state is set to incapable of handling important-labeled messages. The spit context is displayed to the end users and the attack is successful.

4.2 Qualitative and Quantitative criteria

A comparative evaluation of the proposed anti-SPIT mechanisms should define and take into account various qualitative and quantitative criteria. In the following we use the criteria already discussed in [15] and [16] for anti-spit mechanisms evaluation:

Percentage of SPIT calls avoided. This quantitative criterion is an indication of the effectiveness of the mechanism and measures how many SPIT call attempts have been identified and handled by the anti- SPIT mechanism.

Reliability. The precision of making the right adjustments about SPIT calls and callers, in terms of false positive and negative rates. One may measure the ratios of right vs. wrong decisions, false positive vs. false negative. This is again a quantitative criterion and is critical mainly for the spit detection methods, when calls are the subject of examination. Otherwise, when callers are analyzed in terms of their trustworthiness and history, then prevention methods might fail, as well.

Promptitude. Due to the real-time nature of VoIP, quick decisions regarding SPIT detection are a major requirement, especially when legitimate calls are analyzed. In such a case, trustworthy users should not tolerate large delays. This is because spit is mainly bulk transmission of calls, and, thus, a prompt decision might avoid depletion of communication or computing resources. A nimble prevention or detection method minimizes the decision time, that is (a) the time between a call request arrival (e.g., SIP INVITE message) and the acceptance of this calls, or (b) the time between a new entry of a SIP user (i.e., caller) and the determination of his/her rationality.

Human Interference. This qualitative metric represents the transparency of the anti-SPIT mechanisms to the end-user. Normally, the annoyance of the callee should be minimized and, thus, preferences and profiling are essential towards this objective. Thus, when SIP proxies handle and prevent spit, they should include end-user profiling and preferences to their logic as an effort to avoid burden. Profiling might be considered during the end-user registration process.

Resources Overhead for the SIP provider. SIP providers should estimate the required resources for the implementation of the mechanism. This quantitative criterion seems essential for providers, since the number of calls that should be analyzed per unit time might be enormous, whilst registered users might be numerous. It may also include the financial cost of licensing of the anti-spit software, purchasing or leasing of the required hardware and broadband communication lines, customization of the software, training of the responsible staff, and, finally, cost of enhancements, or some in-house development. In some cases, additional service costs should be considered, such as cost per certificate, when PKIs are involved.

Vulnerabilities. This parameter refers to the capability of a spitter to bypass any of the anti-spit countermeasures.

Privacy Risk. This criterion is associated with the collection, manipulation, and dissemination of private data. We assume that the end-user consents for the collection and

manipulation of her private data, and she has authorized specific legal entities for these purposes.

Scalability. This is an important criterion, since VoIP networks grow fast. Scalability should be considered when authentication is involved, since PKI and CA might need to establish complex cross-certification chains, or reputations and assertions are used.

Adoption: The criterion corresponds to the success of an anti-SPIT countermeasure, and depends on the effort it takes for an end-user, or a provider, to begin using it. It may be affected by the effort needed by an end-user to start using the anti-spit mechanism (e.g. installation, learning to use it and performing initial operations) or the effort needed by end-users before the mechanism starts to provide its benefits (e.g. need for configuration).

Availability. It denotes the increase in the availability of network, computing, memory, or human resources when preventing, detecting or handling spit countermeasures apply.

5 Theoretical evaluation of spit avoidance methods

5.1 Fulfilled criteria

The evaluation of the spit avoidance methods is structured according to [15] on the following concept; evaluate which mechanisms measure, estimate or just mention the previously analyzed quantitative and qualitative criteria. In the corresponding research, it has been taken under consideration only the description in the equivalent paper of each mechanism. It must be stressed out that the results of the analysis presented below have not been based on which criteria each mechanism could, should, can or may fulfill or take into consideration. Additionally, it has not been estimated whether the mechanisms meet the criteria well or not, but it is rather provided the existence of each criterion in the mechanisms' description.

For example in the description of VoIP Seal [10], it is mentioned that the particular mechanism is intrusive for the end user, as there is an interactive part requiring user's feedback. In the current analysis, it has not been estimated how intrusive this mechanism is or not, but it has been emphasized the fact that there are information in the description of the mechanism that could help someone value accordingly the particular criterion, namely Human Interference.

On the other hand, in the Network Layer Anti-Spit Entity mechanism [5], it can be inferred from the description of the mechanism, that there is no need for and user involvement during the anti-spit process. However, something like that is not explicitly mentioned in the presentation.

Finally, regarding the vulnerabilities criterion it must be emphasized the following. Perfect security is impossible to deliver especially in open and dynamic environments like the internet. New vulnerabilities appear daily, and spitters will always find ways to overcome any anti-spit countermeasures. Someone can only make harder and profitless the process to achieve their goals. Under this view, in this criterion the research focuses on the vulnerabilities considerations that the authors take into account, whether these are few or many.

Next are presented some comments regarding each mechanism's evaluation result. These points are further summarized in table 1.

In [2], although the mechanism is described in detail no reference is made on any of the previously mentioned criteria. On the other hand, the Network Layer Anti-Spit Entity [5] through the experimental results that it provides, fully covers the Reliability and Percentage of spit calls avoided criteria. Additionally the authors take into consideration the resources required for a SIP provider to employ their mechanism, thus fulfilling the Promptitude and Resources Overhead criteria.

The Reputation/Charging mechanism [6] provides no experimental results, although it mentions that such an evaluation is forthcoming. However no quantitative criteria are met and the analysis of the mechanism does not satisfy any of the qualitative criteria also.

DAPES [7] mechanism's detailed description and provision of various scenarios complies with only one criterion namely vulnerabilities. The absence of experimental results converges to this minimal criteria compliance. The same applies for the DSIP mechanism [4].

RFC4474 [13] is an identity management approach. Although its relevance to Spit has been analyzed in previous sections, the description of the mechanism makes reference to some privacy matters alongside with the security issues of such a method. SIP SAML [8] also covers such topics in relevance to the evaluation criteria, but also cites some Human Interference issues.

Another identity based approach is the Biometrics framework [1]. The authors in their description refer to the issues related to the users' annoyance by the mechanism, and to the security known vulnerability, namely Sybil attack.

The experiments mentioned and presented in [3] fulfil the Percentage of spit calls avoided and Promptitude criteria, whilst the vulnerabilities mentioned correspond to the appropriate criterion.

Finally, VoIP Seal's description [10] indicates information that can only cover the Human Interference criterion. On the other hand, VSD [12] besides fulfilling Human Interference, through the experimental results provided by its authors, covers also the Percentage of spit calls avoided, Reliability and Scalability criteria.

Criteria / **Anti-SPIT Mechanisms**	Percentage of spit calls avoided	Reliability	Promptitude	Human Interference	Resources Overhead for the SIP provider	Vulnerabilities	Privacy Risk	Scalability	Adoption	Availability
AVA	–	–	–	–	–	–	–	–	–	–
Network Layer Anti-Spit Entity	√	√	√	–	√	–	–	–	–	–
Reputation/Charging	–	–	–	–	–	–	–	–	–	–
DAPES	–	–	–	–	–	√	–	–	–	–
PMG	√	–	√	–	–	√	–	–	–	–
Biometrics	–	–	–	√	–	√	–	–	–	–
RFC 4474	–	–	–	–	–	√	√	–	–	–
SIP SAML	–	–	–	√	–	√	√	–	–	–
DSIP	–	–	–	–	–	√	–	–	–	–
VoIP Seal	–	–	–	√	–	–	–	–	–	–
VSD	√	√	–	√	–	–	–	√	–	–

5.2 Categorization, Advantages and Disadvantages

In this section, we provide briefly the advantages and the disadvantages advantages of the techniques described in section 3.1

Technique	Advantages	Disadvantages
[2]. *SPIT prevention using anonymous verifying authorities (AVA)*	▪ Enhances users authentication ▪ Call establishment is considered fast with no noticeable delayed to end-	▪ This solution might not be scalable ▪ It might cause privacy and anonymity problems, according to where the new components are deployed and what are the

▪ Used for SPIT prevention	user.	requirements for the SIP customer ▪ No validation is proposed ▪ Introduces overheads: new hops and when fetching/updating profile
[5]. *SPIT mitigation through a network layer anti-spit entity* ▪ Used for SPIT detection	▪ The proposed detection criteria can be considered as a good start for a more elaborated solution ▪ Fast and efficient ▪ No extra overhead ▪ There are scenarios where false positives can be identified ▪ Increase availability: reducing proxy traffic when filters in use	▪ The technique deployment is questionable ▪ The SPIT classification depends on a certain threshold, however, no performance related to this threshold is provided ▪ The validation part is preliminary, so no decision about the technique accuracy can be made ▪ The convergence might be slow, since it takes some time to identify the enumeration filters, with given criteria.
[6]. *SPIT detection based on reputation and charging techniques* ▪ Used for SPIT detection	▪ Two different techniques are suggested: one based on the reputation concept and the other one is based on a charging scheme ▪ The reputation based technique looks like a combination of white, black and grey lists ▪ Trust and reputation might avoid 100% of spit ▪ The charging based technique suggests a model for collecting and managing the charging information	▪ Only the techniques concept is discussed ▪ No validation so far, so no accuracy decision can be made ▪ The reputation based technique complexity is not clear especially if more than one provider are considered ▪ The charging technique requires agreements between the providers which might be difficult to achieve ▪ When IVR is present, it might create a considerable overhead ▪ Increases availability for end-points, but resources are needed for IVR scheme
[7]. *DAPES* ▪ Used for SPIT prevention	▪ Can prevent spam from open domains if not authenticated ▪ Extremely accurate in false positives when originating from trusted domain ▪ Considered trustworthy due to signed certificates ▪ Reduces unwanted traffic due to certification	▪ This solution might not be scalable due to PKI ▪ Creates overhead because of TCP/TLS

	requirements	
[3]. *Progressive multi-gray levelling* ▪ Used for SPIT detection	▪ Based on user calls history ▪ Very fast and efficient, done through enumeration filters ▪ No extra overhead ▪ Increase availability: reducing proxy traffic when filters in use, but more processing power is needed due to short time and long time traffic analysis ▪ Good for intra-domain security	▪ This solution seems to have problems when using fake identities ▪ Thresholds need to be initially decided ▪ Convergence takes time to identify the enumeration filters, with given criteria ▪ Damage control when loss of credentials
[1]. *Biometric framework for SPIT prevention* ▪ Used for SPIT prevention	▪ This technique bounds the user identity to its personal data ▪ Work efficient under the assumption that every caller is speaking the same language, thus not too feasible to implement	▪ Not necessarily efficient due to potential problems authenticating (voice problems, noise, sensitivity) ▪ Distinguishing between users voices is difficult especially in presence of background noise ▪ The deployment of the authentication servers is questionable ▪ This solution might suffer from scalability problems ▪ Will cause some overhead due to IVR authentication ▪ Will cause overhead due to TLS handshaking ▪ IVR will consume resources in form of bandwidth and customer annoyance
[4]. *DSIP* ▪ Used for SPIT detection and handling	▪ SPIT classification ▪ Blacklisting initiated by the callee will give the callee the option of altering the filters and thus provide consistency ▪ Effectively blocking known spammers will free up resources at the callee home network, but would leave the known spammers to still	▪ It is not clear how white and black lists are built ▪ It seems that some changes in the SIP messages are required ▪ The deployment of this technique is not clear ▪ Human verification tests (for greylisted callers) will necessarily add some substantial delay in the setup ▪ Adds a small overhead besides the

	use of the service providers resources in a backbone network.	considerable amount with human verification tests for VoIP calls.
[13]. *RFC4474* ▪ Used for SPIT prevention	▪ Prevents fake identities	▪ This technique does not do more than verifying that the caller is exactly who is pretending to be, so it needs to be used with other techniques in order to be efficient ▪ Will cause some overhead in certificate negotiation
[10]. *VoIP SEAL* ▪ Used for SPIT detection	▪ Modular approach allowing to respond to any new kind of SPIT and to customize the systems to meet the needs of variety of hardware ▪ The use of two thresholds looks like using a combination of white, black and grey lists ▪ If the detection patterns are at a desirable level, the solution will indeed save resources both in a domain as a whole and at the user endpoints.	▪ A rating system is used for deciding whether a call is SPIT or not. How these thresholds are defined in unclear ▪ The accuracy of the spam detection is directly related to the rate and the quality of which the needed signatures can be provided.
[8]. SIP SAML ▪ Used for SPIT prevention	▪ This technique does not recognize spit, but acts at a deterrent by asserting the identity of the caller ▪ Increase resource availability since tracking spitters (identity) will be easier	▪ Overhead: few new hops introduced will delay establishment, but not considerable ▪ Introducing new entities in the network, holding customer profiles, will be will need maintenance given the detailed per customer information.
[12]. *Voice Spam detector* ▪ Used for SPIT detection	▪ This solution combines various anti-spam techniques which should increase its efficiency ▪ Except potential human errors in the social networks module and blacklistings, the solution itself should	▪ The scalability and the delay that can be caused by the deployment of this solution is not clear ▪ Some techniques seem to overlap ▪ One issue with all Bayesian Learning implementations is that they need a good training set ▪ The framework adds substantial

	provide a fairly good consistency.	costs and need of resources in any practical installation

6 Conclusions

The adoption of VoIP technology and the establishment of SIP as the prevailing signaling protocol introduced significant concerns if the spit affect will be equal to the current spam expansion. VoIP infrastructures have been recently gained a recognizable market share. Thus, only recently, and prior to any spit expansion and risk, some research groups have proposed considerable anti-spit countermeasures. On the other hand, the proposed anti-spit mechanism aims at fulfilling qualitative and quantitative criteria.

In this deliverable we have identify the state-of-the art in the anti-spit mechanisms domain and contacted used a two-fold evaluation framework. First, we defined a set of parameters that each mechanism should address in order to counter SPIT efficiently, and we identified how each class should be evaluated, in terms of effectiveness. Second, we analyzed which of the SPIT identification criteria each SPIT mechanism takes into account. This theoretical evaluation framework provides insight on how the effectiveness of a mechanism can be evaluated and how combinations of mechanisms can be selected in order to effectively mitigate SPIT in a given context.

7 References

[1] R. Baumann, S. Cavin, and S. Schmid, “Voice Over IP - Security and SPIT”, Swiss Army, FU Br 41, KryptDet Report, University of Berne, August 24 - September 8, 2006

[2] N.J. Croft, and M.S. Olivier, "A Model for Spam Prevention in Voice over IP Networks using Anonymous Verifying Authorities," in Proc. of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June/July 2005

[3] S. Dongwook, A. Jinyoung, and S. Choon, “Progressive Multi Gray-Leveling: A Voice Spam Protection Algorithm”, IEEE Network, 20 (5), pp. 18-24, September/October 2006

[4] Madhosingh, “The Design of a Differentiated SIP to Control VoIP Spam”, Technical Report, Computer Science Department, Florida State University, 2006

[5] B. Mathieu, Q. Loudier, Y. Gourhant, et al., “SPIT Mitigation by a Network-Level Anti-Spit Entity”, in Proc. of the 3rd Annual VoIP Security Workshop, June 2006, Berlin, Germany.

[6] Y. Rebahi, D. Sisalem, and T. Magedanz, T.; “SIP Spam Detection”, in Proc. of the International Conference on Digital Telecommunications, pp. 29-31, Aug. 2006, France.

[7] K. Srivastava, and H. Schulzrinne, “Preventing Spam For SIP-based Instant Messages and Sessions”, Technical Report, University of Columbia, 2004.

[8] H. Tschofenig, R Falk, J. Peterson, et al., “Using SAML to Protect the Session Initiation Protocol (SIP)”, IEEE Network, 20 (5), pp. 14-17 September/October 2006.

[9] Z. Duan, Y. Dong, and K. Gopalan, "DMTP: Controlling Spam Through Message Delivery Differentiation", in Proc. of Networking Conference 2006, Coimbra, Portugal, May 15-19, 2006.

[10] S. Niccolini, “SPIT prevention: state of the art and research challenges”, Network Laboratories, NEC Europe Ltd., Heidelberg, Germany.

[11] S. Niccolini, R. G. Garroppo, S. Giordano, et al., “SIP intrusion detection and prevention: recommendations and prototype implementation”, in Proc. of 1st IEEE Workshop on VoIP Management and Security, pp. 47 – 52, 2006.

[12] R. Dantu, and P. Kolan, “Detecting Spam in VoIP Networks”, in Proc. of Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI ‘05), July 2005, Cambridge, MA, USA

[13] RFC4474, J. Peterson, and C. Jennings, “Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)”, August 2006

[14] Eleven GmbH, January 2007, http://www.eleven.de/print/en/company/news/graphic/?plain=1

[15] S. Dritsas , J. Mallios, M. Theoharidou, G. F. Marias, and D. Gritzalis, “SPIT Identification Criteria and Anti-SPIT Mechanisms Evaluation Framework”, submitted on IEEE Global Telecommunications Conference (IEEE GLOBECOM 2007), Washington, D.C., U.S., Nov. 2007

[16] G. F. Marias, S. Dritsas, M. Theoharidou, J. Mallios, and D. Gritzalis, “SIP Vulnerabilities and Anti-Spit Mechanisms Assessment”, submitted on 16th International Conference on Computer Communications and Networks (IEEE ICCCN 2007), Honolulu, Hawaii, US, Aug. 2007