

The Design of a Differentiated Session Initiation Protocol to Control VoIP Spam

By

ADRIAN RISHI MADHOSINGH

A PROJECT PRESENTED TO THE GRADUATE SCHOOL  
OF FLORIDA STATE UNIVERSITY IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF COMPUTER SCIENCE

FLORIDA STATE UNIVERSITY

2006

Copyright 2006

by

Adrian Madhosingh

## ACKNOWLEDGMENTS

I would like to thank my family for their support. I would like to thank Dr. Zhenhai Duan for his commitment and inspiration to helping me achieve my goal. Dr. Zhenhai Duan has provided me with essential advice and extraordinary guidance that helped me express my ideas. I would like to thank Dr. Kartik Gopalan and Dr. Breno de Medeiros for all their efforts including the time and effort required to serve on this committee. Their insight and classroom teaching is exceptional and has a profound impact on me.

## TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS .....	iii
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
ABSTRACT .....	viii
CHAPTER	
1 INTRODUCTION .....	1
Background and Motivation .....	1
Contribution .....	2
DSIP Overview .....	3
DSIP Advantages .....	3
Structure of Paper .....	4
2 OVERVIEW OF VOIP AND SIP .....	5
History of VoIP .....	5
Organizational Involvement .....	5
Centralized versus Decentralized Architectures .....	6
Centralized Architecture .....	6
Decentralized Architecture .....	6
Comparison .....	6
Session Initiation Protocol (SIP) .....	8
SIP Main Components .....	9
User Agent .....	9
Servers .....	10
SIP Message Types .....	10
Requests .....	10
Responses .....	11
3 CLASSIFICATION ROUTING PHASE .....	13
Caller Classification .....	13
Whitelisted Classification .....	13

Blacklisted Classification .....	14
Greylisted Classification .....	14
Verified Classification .....	14
Classification-based Call Routing .....	15
Caller Classification Considerations.....	17
4 HUMAN VERIFICATION PHASE.....	19
An Example Human Verification Test .....	19
Message Formats .....	22
5 VOICEMAIL NOTIFICATION PHASE .....	24
6 VOICEMAIL RETRIEVAL PHASE .....	30
7 SUMMARY.....	34
<b>APPENDIX</b>	
A CLASSIFICATION ROUTING IMPLEMENTATION .....	36
Code Insertion.....	38
proxy.c .....	38
siproxd.c .....	40
Routing Verification .....	41
Whitelist Example .....	42
Greylist Example .....	42
Blacklist Example.....	42
Verified Example.....	43
LIST OF REFERENCES .....	44

## LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 SIP Response Code Categories .....	12
A-1 RFC3261 Request Processing Checklist .....	36
A-2 RFC3261 Request Validation Checklist .....	37
A-3 DSIP Request Validation Checklist .....	37

## LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 Overview of a centralized architecture.....	7
2-2 Overview of a decentralized architecture.....	8
2-3 Overview of SIP protocol.....	9
3-1 Example DSIP classification routing phase blacklisted request message.....	15
3-2 Flowchart of DSIP classification routing phase.....	17
4-1 SIP call flow of the human verification process .....	20
4-2 Example of normal BYE request .....	22
4-3 Example DSIP human verification phase approval message .....	23
4-4 Example DSIP human verification phase rejection message .....	23
5-1 SIP Call flow of the voicemail notification phase.....	25
5-2 Example DSIP voicemail notification phase voicemail notification request message .....	26
5-3 Example DSIP voicemail notification phase voicemail notification message .....	28
6-1 SIP Call flow of the voicemail retrieval phase.....	31
6-2 Example DSIP voicemail retrieval phase voicemail retrieval request message.....	31
6-3 Example DSIP voicemail retrieval phase voicemail retrieval data message.....	32

Abstract of Project Presented to the Graduate School  
of Florida State University in Partial Fulfillment of the  
Requirements for the Degree of Master of Computer Science

THE DESIGN OF A DIFFERENTIATED SESSION INITIATION PROTOCOL TO  
CONTROL VOIP SPAM

By

ADRIAN RISHI MADHOSINGH

Chair: Dr. Zhenhai Duan

Committee Members: Dr. Kartik Gopalan, Dr. Breno de Medeiros

Major Department: Computer Science

The explosion of the Internet has generated a wide array of technological advances related to packet-switched data networks. As a result it is becoming more reasonable for businesses and consumers to communicate through their internet connections as opposed to their traditional telephone carrier networks. VoIP technologies take advantage of existing data networks to provide inexpensive voice communications worldwide as a promising alternative to the traditional telephone service. However, one important aspect of the VoIP boom is largely overlooked, namely the threat of VoIP spam. Spam over Internet Telephony (SPIT) not only could be a major annoyance in the future, it has the realistic potential to cripple the growth of VoIP, and thus the usefulness of the cost effective technology.

This project proposes a differentiated session initiation protocol (DSIP) to control VoIP spam. DSIP extends the original SIP protocol to include mechanisms to control SPIT. In DSIP, a callee can classify callers into three classes---whitelist (regular

contacts), blacklist (known VoIP spammers), and greylist (neither regular contacts nor known spammers), and handle the corresponding communication differently. For example, while callers in the whitelist can directly communicate with the callee using the current SIP protocol, callers in the blacklist class are not allowed to communicate at all. Callers in the greylist must pass a human verification phase to communicate with the callee. Importantly, callers in the greylist, after passing the human verification phase, still cannot leave voicemail messages in the callee's voicemail server; such callers must store the voicemail messages on their own voicemail servers. The design of DSIP is heavily influenced by the Differentiated Mail Transfer Protocol (DMTP) to control email spam.

DSIP has many advantages with respect to SPIT control. First, DSIP provides end users with greater flexibility over who can contact them, and mitigates voicemail denial of service attacks. Second, DSIP helps to hold VoIP spammers responsible for spamming. Lastly, DSIP imposes legitimate callers and recipients with little to no overhead in their normal SIP transactions and multimedia communications, while penalizing abusive callers with the expense of disk space and CPU cycles, in an effort to globally mitigate the onset of SPIT and voicemail denial of service attacks. With low impact on the current SIP specification, the DSIP is a valuable modification with a low cost to benefit ratio that will benefit the growth of VoIP communications into the future.

## CHAPTER 1 INTRODUCTION

### **Background and Motivation**

The explosion of the Internet has generated a wide array of technological advances related to packet-switched data networks. Development in data networks has occurred more quickly than the traditional telephony circuit switched network. As a result, and with the growth of broadband connectivity, it is becoming more reasonable for businesses and consumers to communicate through their Internet connections as opposed to their traditional telephone carrier networks. VoIP technologies take advantage of existing data networks to provide inexpensive voice communications worldwide as an alternative to the traditional telephone service. There are many advantages to VoIP including cost savings, open standards, and integrated networks. VoIP applications provide cost savings as they reduce their reliance on the Public Switched Telephone Network (PSTN) and only require maintenance of one network of intermixed voice and data.

With the explosion of broadband subscription among consumers, both business and residential users are able to take advantage of VoIP technologies. The Telecommunications Industry Association reports that the U.S. broadband market has grown from 4.5 million subscribers in 2000 to 41.3 million subscribers in 2005 with expectations of growth to 69.2 million by 2009 [G06]. The Telecommunications Industry Association also reports that the number of residential VoIP customers more than tripled to 4.2 million in 2005 and is expected to grow to 18.0 million by 2009, with revenues jumping from \$25 million in 2003 to \$1.1 billion in 2005, and with projections of \$5.1

billion by 2009 [G06]. It is clear that VoIP communication is growing at an extremely fast pace, with the low cost of broadband services and competition driving the growth.

One overlooked aspect of the VoIP boom is the threat of VoIP spam. Spam over Internet Telephony (SPIT) not only could be a major annoyance in the future, it has the realistic potential to cripple the growth of VoIP, and thus the usefulness of the cost effective technology. SPIT can be found in the form of unwanted calls and voicemail, and can be expanded to more serious denial of service attacks. SPIT has the potential to be more malicious than email spam, due to the fact that it requires users to analyze if a call or voicemail is useful or wanted. Telephony, by nature, requires immediate attention of the user in real time linear fashion, as opposed to spam, which can be collected and analyzed in an email inbox intermediately over a period of time. Voice communication requires the immediate and sole attention of the user, and repeated interruption, in the form of SPIT, can be devastating to productivity. Unwanted voicemail can impair productivity because the attention and time required to listen to a voicemail far exceeds that of its email spam counterpart. If left uncontrolled SPIT will be more than annoying, SPIT will also have an enormous economic impact as it will drain the productivity of users. SPIT is currently difficult to stop due to the nature of anonymity in the Internet. It is important that any user should be able to contact another user, but because callers constantly change locations and Internet connections, it becomes difficult to isolate and uniquely identify a caller.

### **Contribution**

This project designs a differentiated session initiation protocol (DSIP) to control VoIP spam. Simply stated, DSIP intends to reduce the amount of non-legitimate communication while not interrupting the normal call flow of legitimate communications.

DSIP is a modification of the SIP specification and extends SIP's feature set to include internal mechanisms to identify and mitigate the onset of SPIT. The design of DSIP is heavily influenced by the Differentiated Mail Transfer Protocol (DMTP) proposed by Duan, Dong, and Gopalan to control email spam [DDK06].

### **DSIP Overview**

DSIP accomplishes the task of screening SPIT by classifying incoming callers with each classification of caller having different sets of privileges. In DSIP, a callee can classify callers into one of three classes---whitelist (regular contacts), blacklist (known VoIP spammers), and greylist (neither regular contacts nor known spammers) and handle the corresponding communication differently [DDK06].

- 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 human verification phase, 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 DMTP [DDK06].

### **DSIP Advantages**

DSIP has many advantages with respect to SPIT control. First, DSIP provides end users with greater flexibility over who can contact them, and mitigates voicemail denial of service attacks. Second, DSIP helps to hold VoIP spammers responsible for spamming. And lastly, DSIP imposes legitimate callers and recipients with little to no

overhead in their normal SIP transactions and multimedia communications, while penalizing abusive callers with the expense of disk space and CPU cycles, in an effort to globally mitigate the onset of SPIT and voicemail denial of service attacks.

With low impact on the current SIP specification, DSIP is a valuable modification with a low cost to benefit ratio that will benefit the growth of VoIP communications into the future.

### **Structure of Paper**

Chapter 2 will give an overview and introduction to VoIP technology and the SIP specification. Chapter 3 will introduce the caller classification of DSIP and how calls are routed based on classifications. Chapter 4 will discuss the human verification mechanism of DSIP, which will be imposed on greylisted callers. Chapter 5 will discuss the voicemail notification mechanism of DSIP. Chapter 6 will discuss the voicemail retrieval mechanism of DSIP, and finally Chapter 7 will summarize the DSIP project.

## CHAPTER 2 OVERVIEW OF VOIP AND SIP

In this chapter, we present a brief overview of the VoIP technology and the Session Initiation Protocol (SIP) to initiate the voice communication on the Internet. We focus on the aspects that are most relevant to our project.

### **History of VoIP**

VoIP started to be commercialized in 1995. However there was a lack of standardization for VoIP and as a result there were four prominent implementations of VoIP: H.232, H.248, Media Gateway Control Protocol (MGCP), and Session Initiation Protocol (SIP). It was up to consumers to pick and choose which standardization they wanted. Each of these protocols has their own strengths and weaknesses, but VoIP would benefit from one protocol implementation. This would reduce dependencies and proprietary equipment, and promote interoperability.

### **Organizational Involvement**

There are two main organizations involved in the protocols and standards of VoIP. The Internet Engineering Task Force (IETF) is an organization that concerns itself with the evolution of the Internet, its protocols, and standards. The International Telecommunication Union is an international organization that works with government and the private sector to coordinate telecommunication networks and services. These two organizations have worked independently and together to provide recommendations for architectures for creating multimedia applications, including VoIP.

## **Centralized versus Decentralized Architectures**

### **Centralized Architecture**

The centralized architecture was the traditional means of delivering VoIP. The centralized architecture allowed for greater control of components including call agents and end points. This results in simplified management but stifles innovation. The protocols that implement a centralized architecture include H.248 and MGCP.

### **Decentralized Architecture**

The decentralized architecture is very similar to that of an IP network and implements features between endpoints as opposed between centralized servers. These features can range from states of calls to routing and call handling. The downside to this architecture is a more complex management. The protocols that implement a decentralized architecture include H.323 and SIP.

### **Comparison**

As mentioned before the main tradeoff between architectures are easy of management and innovation. Fortunately, VoIP can be implemented in either flavor in terms of a centralized or decentralized architecture based on the needs of the end users.

In a centralized environment the end points communicate with their local call agent. Then the call agent communicates with other call agents to eventually deliver the data to the other intended endpoint.

This process is almost identical to the way that email is delivered in that a local client program communicates with its local mail transfer agent. The local mail transfer agent then communicates with the intended recipient's mail transfer agent in which the message is the delivered locally.

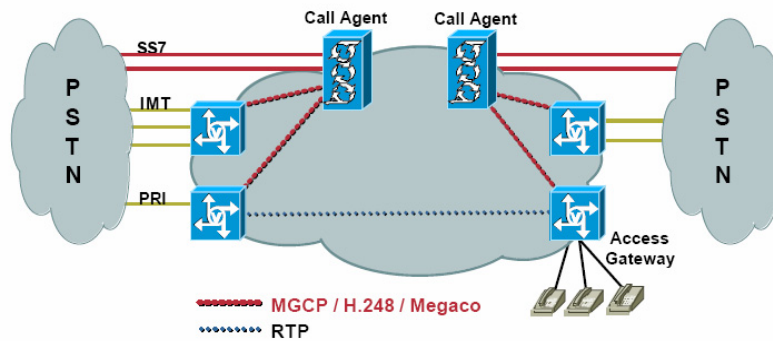


Figure 2-1. Overview of a centralized architecture [CISCO].

The drawback to this system is that as features are implemented, the call agents must be updated to support new features. If two endpoints want to use a new feature both call agents must be updated, which may be difficult to do and/or expensive.

Advocates of a centralized architecture are in favor of simplicity and implementation of basic features. This architecture makes it difficult for features to be standardized and discourages developers from even attempting to develop new features.

A decentralized architecture is much more powerful than centralized architecture and provides more room for innovation. Instead of endpoints being dumb terminals, as in a centralized architecture, all features are implemented at the endpoints. When endpoints initialize a conversation, they can synchronize their available features for use. This eliminates the dependency of call agents being up to date, thus saving time and money.

In a decentralized architecture, and SIP specifically, there does exist a centralized server. However, the purpose of this centralized server is call registration and the initialization of communication of endpoints.

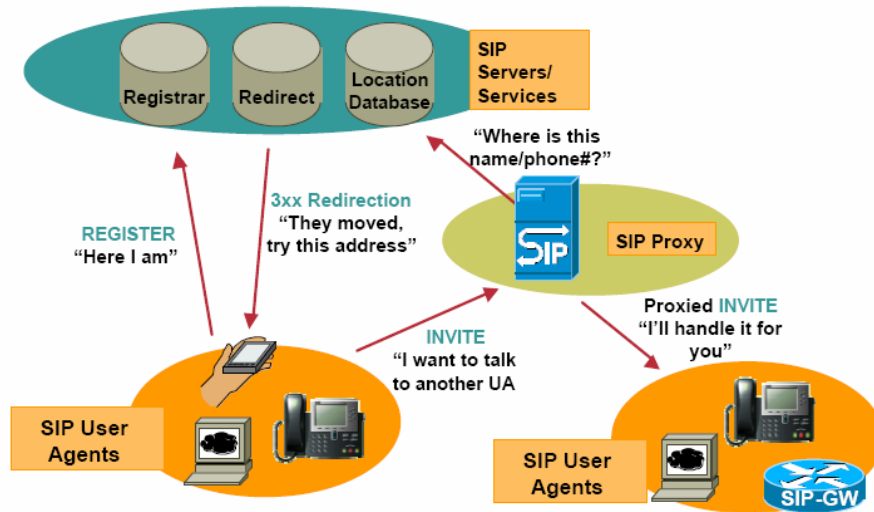


Figure 2-2. Overview of a decentralized architecture [CISCO].

The benefits of a decentralized architecture allow for a greater range of flexibility and allows for end users to choose the features at their endpoints instead of relying on the capabilities of a centralized call agent.

### Session Initiation Protocol (SIP)

There already exist many protocols that carry many different of real time multimedia session data on the Internet. The Session Initiation Protocol (SIP) works in conjunction with these protocols to help end points (user agents) discover one another and agree on the details of the type of session that they would like to share.

SIP enables the creation of a 3rd party proxy server that serves as the central communication point for registrations, invitations, and many other requests. SIP works independent of the transport layer and does not have a dependency on the type session that end point will use to communicate.

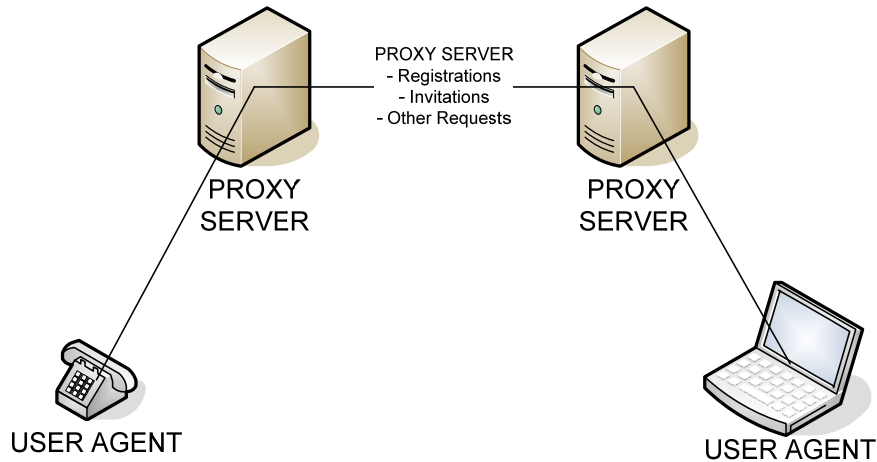


Figure 2-3. Overview of SIP protocol.

SIP is an application layer protocol that is used to create, maintain, and destroy multimedia sessions. There are five main components to multimedia communication that SIP supports: User location, User availability, User capabilities, Session setup, and Session management. User location deals with the locations of user agents. User availability deals with the willingness of user agents to communicate with each other. User capabilities deal with the negotiation of session parameters among the user agents. Session setup deals with the establishment of the actual connection. And finally, session management deals with the teardown and transferring connections, modifying the session parameters, and controlling the services of the session.

### **SIP Main Components**

#### **User Agent**

User Agents (UA) are endpoints for communication. This can include actual SIP phones or computers equipped to handle voice communication (soft phones). At the most basic level these devices can function as an User Agent Client (UAC) or an User Agent Server (UAS), with the difference being that a UAC only initiates a request as an UAS contacts the user (rings) when a request is received and automatically replied to.

We can think of an UA acting as a UAC when we pick up the phone to make a call. And we can think of an UA acting as a UAS when the phone rings because someone has sent a request. It is the case that when the UA is in use it is acting as either a UAC or UAS but not both at the same time.

Another type of SIP clients are gateways that serve as a gateway for other services including protocol translation. A gateway may also function as a call manager, which provides voicemail services or call routing.

### **Servers**

There are three types of servers: proxy, redirection, and registrar. The proxy server serves as the centralized midpoint of communication that facilitates the location and setup of UACs and UASs. Typically, it receives invitation requests from UACs and sends them to the appropriate UAS. Proxy servers can also be the centralized point for security, routing, and QoS. Redirection servers simply provide information as to the next hop that a message needs to make to reach its destination. Registration servers act as a centralized location for information about UACs and UASs locations. Normally these are situated in a place that makes it easy for a proxy server to communicate and retrieve information about UAs.

### **SIP Message Types**

There are two types of messages in SIP. SIP messages are text based, and are either of a request or response type. The format of the message is RFC2822 compliant and with exceptions to syntax and the adaptation of header fields.

### **Requests**

There are six types of requests described as methods. These methods include REGISTER, INVITE, ACK, CANCEL, BYE, and OPTIONS.

The REGISTER request can be used to add, remove, or query bindings. As part of the registration process a user agent sends a REGISTER request to a user agent server (UAS). This REGISTER request can aid in the construction, modification, or removal of a mapping between an address-of-record and a user's contact address. The INVITE request is the UA's method of invoking a session. The UA sends an INVITE request to the UAS for potential acceptance and initiation of a session. The ACK request has the most stringent requirements of the requests and can have the least amount of variability. Simply stated, the ACK request is used for confirming receipt of a message. The CANCEL request only cancels pending transactions. Once a final response has been sent a CANCEL request will not affect the transaction.

The BYE request is used to abandon sessions. Within a two-person session, a BYE request would implicitly terminate the entire session. In a multi-user session with more than two people, a BYE request would not terminate the session but would rather let a user "drop off" from the session. In practice, when a user "drops off" a session they do not send a BYE request but rather just terminate their session. The OPTIONS request allows for a UA to query another UA or proxy server for available compatibilities. Information related to methods, content-types, etc, can be queried without sending an INVITE request.

## **Responses**

A reaction to a request is called a response. Each type of response is associated to a number ranging from 100-699. These response codes extend HTTP/1.1 response codes generally, although some are not relevant to SIP. These response codes are divided by the hundredth digit with each range representing a broad overview of the type of response.

1xx response codes are informational  
2xx response codes are services related  
3xx response codes pertain to redirection  
4xx response codes pertain to client error  
5xx response codes pertain to server error  
6xx response codes pertain to global failures

Table 2-1. SIP Response Code Categories

## CHAPTER 3 CLASSIFICATION ROUTING PHASE

As we have discussed in the first chapter, a callee in the Differentiated Session Initiation Protocol (DSIP) can classify callers into three classes and differentiate the voice communication with a caller depending on the class the caller belongs to. In this chapter we first discuss the caller classification in detail and then we describe the corresponding call routing for each class in DSIP.

### **Caller Classification**

Callers are classified into three classes, whitelist, blacklist, and greylist [DDK06]. And moreover, callers in the greylist class are temporarily classified into a verified class for special handling, after they pass the human verification phase. In the following, we discuss the three classes.

#### **Whitelisted Classification**

The whitelisted classification is defined as being a trusted caller or regular contacts of a specific callee. Callers in the whitelisted class can directly communicate with the callee, and leave voicemail messages on the callee's voicemail server when the callee is not available for the call. Put in another way, calls from whitelisted callers are handled in the same way as in the current SIP based VoIP system. There is no extra burden for regular correspondence using the DSIP protocol. The local database of whitelisted users should be maintained on the callee's proxy server.

**Blacklisted Classification**

Callers in the blacklisted class are known VoIP spammers or the ones that the callee does not wish to communicate with. Calls from the blacklisted callers are directly rejected by the callee's proxy server. The local database of blacklisted users should be maintained on the callee's proxy server.

**Greylisted Classification**

Callers in the greylisted class are the ones that belong to neither whitelisted class and the blacklisted class. They are normally the first time callers and the ones that have not been classified. A callee in general has less trust in greylisted callers. Such callers need to pass a human verification test to ensure with a high probability that a call is initiated by a human being instead of an automated caller. After a greylisted caller passes the human verification test, he is temporarily classified into a verified class (see below). Otherwise, if a caller fails the test, the call is directly rejected by the callee's proxy server.

**Verified Classification**

The verified classification contains the greylisted callers who have passed the human verification test. Each entry in the verified class is associated with a timer. When the timer expires, the corresponding caller is removed from the verified class. Such caller will remain in the greylisted class, unless the callee explicitly adds it to the whitelisted or blacklisted class. Callers in the verified class can directly communicate the callee. When the callee is not available, however, the callers cannot leave a voicemail message in the callee's voicemail server. Instead, they have to store the message on their own voicemail server and inform the callee how the message should be retrieved, in the same way how unclassified senders send email in DMTP [DDK06]. It is under the control of the callee if

and when the message will be retrieved. It should be up to the callee to add trusted callers to the whitelisted or blacklisted classification.

### **Classification-based Call Routing**

When an INVITE request is received by the callee's proxy server it forwards the message based on classification. If the caller is classified as whitelisted, then the SIP call flow will proceed as normal between the caller and caller.

If the caller is classified as blacklisted then the SIP call flow will immediately end with a modified BYE request. In a normal SIP call flow the callee is expected to maintain state after sending the BYE request, but in this case the proxy server will not expect or process anymore SIP messages for this particular call flow identified by the "CallID" header. Points of interest include the "ContentLength" header, which describes the length of the message body and indicates that no message body content. For the blacklisted rejection message, it will be necessary to modify this "ContentLength" header and insert the "ContentType" header to describe the message body. The first and only line of the message body will indicate "BLACKLISTED" to indicate to the caller that they were unsuccessful in the classification routing phase.

```

BYE sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
;received=192.0.2.222
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
;received=192.0.2.201
Max-Forwards: 68
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 12
Content-Type: text/plain
BLACKLISTED

```

Figure 3-1. Example DSIP classification routing phase blacklisted request message

If the caller is classified as verified, the callee's proxy server will check its local database to verify that the caller has a valid verified classification. It will verify that the caller exists in the verified database and that the time frame that was passed on to the caller during the human verification phase has not elapsed. If the caller does not have a valid verified classification the caller will be treated as a greylisted caller and subsequently passed onto the human verification phase. If the caller does have a valid verified classification status then the SIP call flow will proceed as normal and the INVITE request will be forwarded onto the desired callee. Finally, if the caller is classified as greylisted or does not exist in any local database the caller will be passed to the human verification phase.

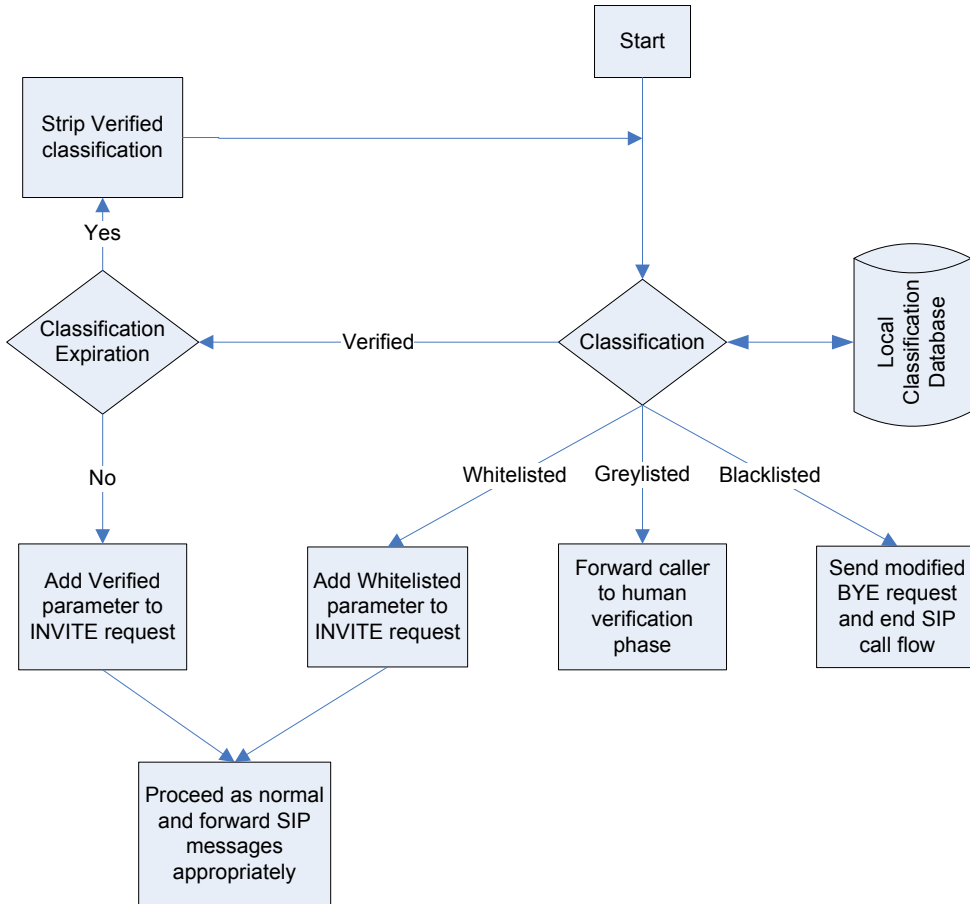


Figure 3-2. Flowchart of DSIP classification routing phase

### Caller Classification Considerations

DSIP identifies callers using their URI. This URI, specified in SIP call set up messaging, is very similar to an email address in structure and consists of two components: a username and domain in the form of `username@domain`. DSIP recognizes the possibility of impersonation and spoofing and therefore the caller identification is important to properly identify and classify callers.

The domain portion of a URI is verifiable through multiple mechanisms. As an example, the callee, could verify that incoming request came from the domain, by cross referencing the information provided with information available. This example, is similar

to a reverse DNS lookup in verifying a hostname to ip address mapping. If a caller relays an INVITE request through its proxy server on its way to a callee, and claims to be user@example.com, then the callee's proxy server can attempt to find out what is the proxy server for example.com domain. If this information can be cross references and matches up correctly then can be properly identified. It is however possible that a user can impersonate another user from the same domain. This provided mechanism is only one example of a scheme that can be used to properly identify callers prior to classification and other schemes could be substituted in this place.

## CHAPTER 4 HUMAN VERIFICATION PHASE

In this chapter we discuss the human verification test imposed on callers in the greylisted class. The test determines if a greylisted caller is a human or automated caller with a desired probability. Because it is highly likely that automated callers will be calling with SPIT, it is important to filter and reject automated callers. First we will present a simple human verification test, and then we describe the message formats to convey the test result to the caller.

### **An Example Human Verification Test**

The purpose of the human verification phase is to determine whether a caller is a human or an automated machine. This can be determined in a variety of ways, but the common goal is to accurately determine if a caller is human or automated machine with such a mechanism that is not intrusive and irritation free. In modern times we see mechanisms to verify users as humans in a variety of ways, including biomedical mechanisms (e.g. fingerprint scanners), picture verification (e.g. user shown picture and asked to identify contents, such as a string of text), and email verification (e.g. user is asked to respond to a confirmation email). All of these mechanisms have their strengths and weaknesses but are not suitable in a VoIP environment.

The following implementation is only an example, whereas any other identification scheme could be easily substituted in this place. DSIP uses a mechanism which the caller is read a set of instructions aloud, via an RTP session, and asked to repeat the string back using the touch tones on the phone, over the RTP session. It is important to note that this

method of human verification is only one of many. Tones from a telephone are already standardized and identified as dualtone multifrequency digits (DTMF). Verification of caller tones can be done over an established RTP session by listening for the RTP payload of a specific DTMF digits, as specified in RFC2833.

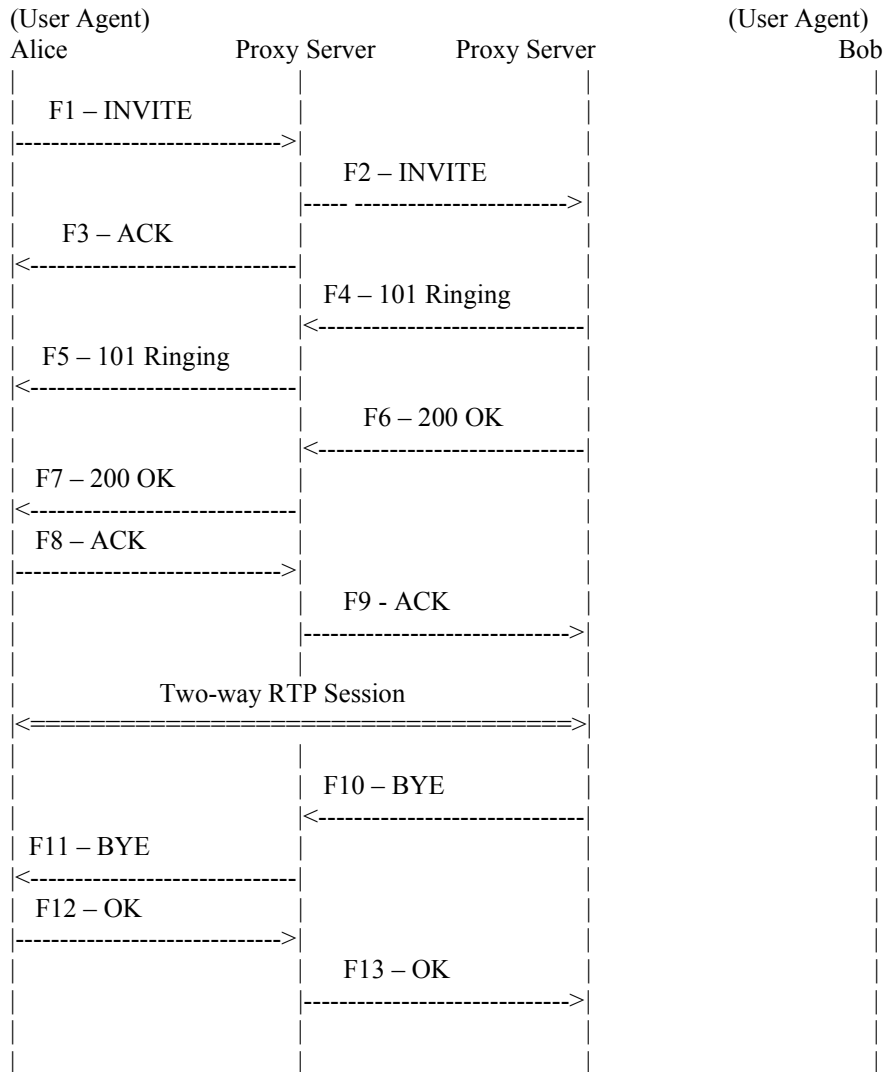


Figure 4-1. SIP call flow of the human verification process

Based on the normal SIP trapezoid, the human verification phase is an interaction between the caller and callee's proxy server. The callee's proxy server would actually

setup an RTP session with the caller so that verification can occur without bother to the callee. The actual verification occurs over the RTP session. As the callee's proxy server will read aloud a string, the caller can then interpret and duplicate the string with tones which are sent back to the callee's proxy server over the RTP session. The callee's proxy server will determine the correctness of the input and will send back an approval or rejection message to the caller. This response is however not sent over the RTP media session, but instead piggybacked with the normal SIP messaging that is required to teardown the established RTP media session. More specifically, the final BYE requests sent from the callee's proxy to the caller will include vital information about the human challenge.

There are two different types of messages that can be sent back to the caller, dependent on the success of the human verification test. If the caller is determined to be human, the caller will be sent back an approval message along with a time frame which describes the length of time for which the approval is valid. This time frame of validity is determined by callee's local policy. Under normal circumstances, the caller will immediately reattempt to connect to the caller after receiving an approval. The caller's proxy server will automatically reattempt this request on behalf of the caller. This automation will prevent the need for the caller to have to redial and spare the caller of any inconvenience. The caller can be assured that they will be forwarded on to the caller if they retry within the approval's time frame of validity. If the caller cannot be verified as a human, the caller will be sent back a rejection message along with a time frame, which describes the length of time for which the caller should wait before reattempting to

connect to callee. The length of the reconnect timeout time frame is determined by callee's local policy.

### Message Formats

Both the approval and rejection messages are to be piggybacked on the BYE request sent from the callee's proxy to the caller during the normal tear down of the established RTP session. In a normal SIP transaction, this request does not have any message body in the actual SIP message.

```

BYE sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
;received=192.0.2.222
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
;received=192.0.2.201
Max-Forwards: 68
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 0

```

Figure 4-2. Example of normal BYE request

Points of interest include the “ContentLength” header, which describes the length of the message body and indicates that no message body content. For both approval and rejection messages, it will be necessary to modify this “ContentLength” header and insert the “ContentType” header to describe the message body. The first line of the message body will indicate either “APPROVE” or “REJECT” to indicate whether or not the caller was successful in the human verification phase. The second line of the message body will indicate a length of time in seconds. If the user was successful in the human verification phase, this time will represent the length of time for which the approval is valid for, and if the user was unsuccessful during the human verification phase this length of time will

represent the amount of time that the caller must wait before reattempting to connect to callee.

```

BYE sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
;received=192.0.2.222
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
;received=192.0.2.201
Max-Forwards: 68
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 12
Content-Type: text/plain
APPROVE
120

```

Figure 4-3. Example DSIP human verification phase approval message

```

BYE sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
;received=192.0.2.222
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
;received=192.0.2.201
Max-Forwards: 68
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 12
Content-Type: text/plain
REJECT
120

```

Figure 4-4. Example DSIP human verification phase rejection message

## CHAPTER 5 VOICEMAIL NOTIFICATION PHASE

Recall that a caller in the verified class cannot leave a voicemail message on the callee's voicemail server. Instead, the caller has to store the message on his own voicemail server. In this chapter we present the mechanism for the caller to inform the callee how such voicemail messages should be retrieved if the callee wishes to do so.

The voicemail notification phase is charged with the task of notifying the caller that the callee is not available. In principle, it is identical to the function provided by the MSID (message id) command in DMTP [DDK06]. If the caller would like to store a media file with a message, they can provide the details to the callee for message retrieval at callee's convenience. This voicemail notification phase is only required for verified classified callers. If the caller is classified as whitelisted, then the normal mechanisms for leaving voicemail is employed. Normally a RTP session is established with the local voicemail system. If the caller is verified then the caller must store the media file for retrieval at later time by the callee.

The first step of the voicemail notification phase is for the callee to notify the caller that the callee is not available, and that if they would like to leave a message they can provide details for voicemail retrieval. As with the human verification phase, there will be no need to add an additional SIP request and respond messaging to the SIP specification to implement this schema. Instead, as with the human verification phase, the necessary information can be piggybacked with normal SIP session tear down messages.

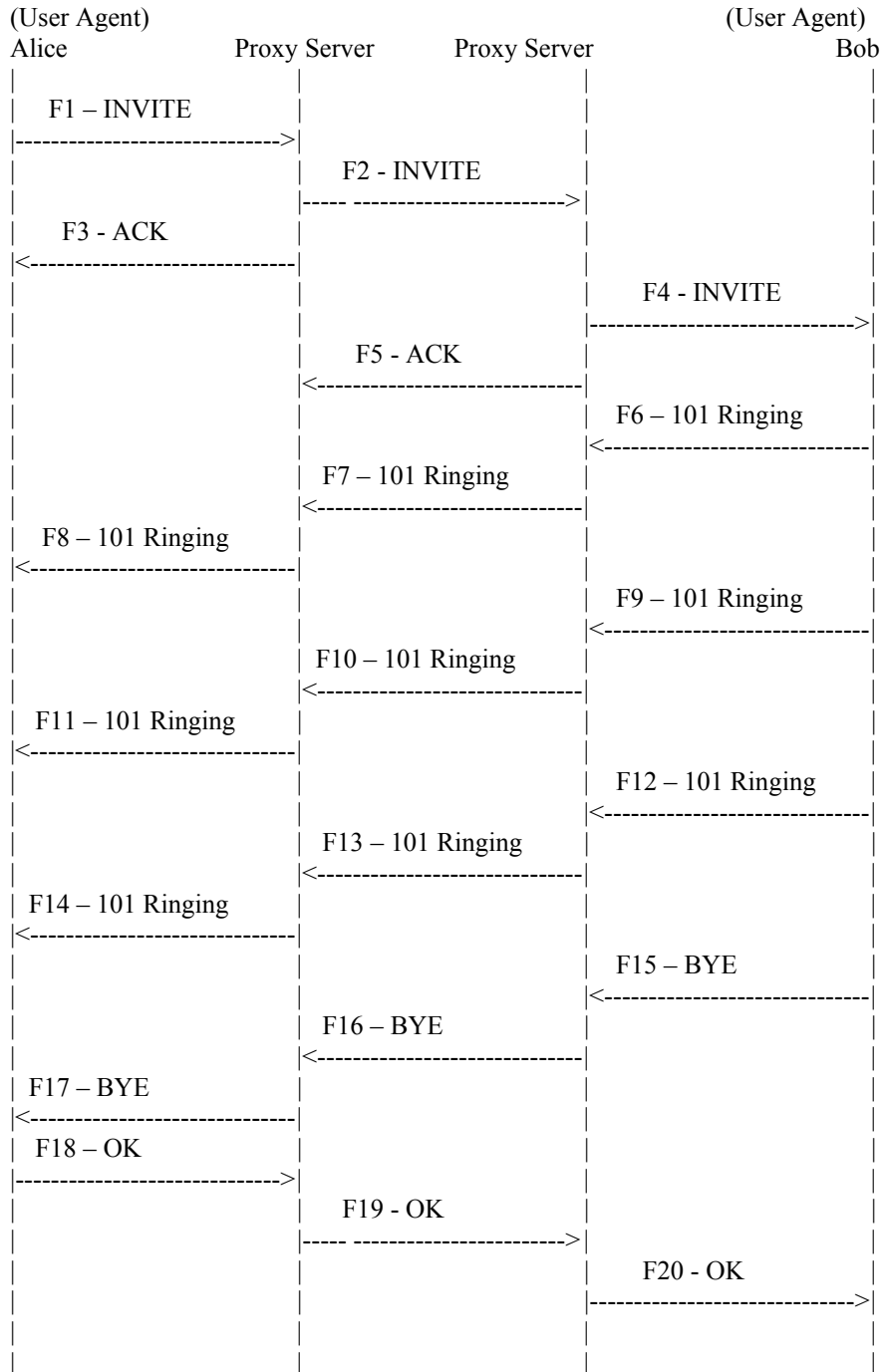


Figure 5-1. SIP Call flow of the voicemail notification phase

After the callee has been deemed not available, as determined by local policy (violation of TRYING response threshold), the callee will send a BYE request to the caller. The information that will be sent back to the caller include the public key of the

callee along with a time frame. The time frame that is transmitted back to the caller will represent the amount of time that the caller will have to transmit back the information required to retrieve the message. This expiration time is required to allow caller enough time to locally record message, perform cryptographic functions, and transmit message to callee. This time frame that is sent to caller can be adjusted according to local policy and can be use to prevent excessively long voicemail messages. Based on the size of the public key transmitted to the caller, the callee can estimate the length of time that will be required to encrypt a message.

```

BYE sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
;received=192.0.2.222
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
;received=192.0.2.201
Max-Forwards: 68
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 BYE
Content-Length: 12
Content-Type: text/plain
VOICEMAIL NOTIFICATION REQUEST
HKDJ9335K3H45K3H45K34H5K3H5KJ87893H4T8DKWHFLDSFJSLVJOSIGJO34
RSA/128-bit
2500

```

Figure 5-2. Example DSIP voicemail notification phase voicemail notification request message

Points of interest include the “ContentLength” header, which describes the length of the message body and originally indicated that no message body content existed. For the voicemail notification request message, it will be necessary to modify this “ContentLength” header and insert the “ContentType” header to describe the message body. The first line of the message body will indicate “VOICEMAIL NOTIFICATION

REQUEST” to indicate to the caller that the callee is not available, and that if they would like to leave a message they can send back retrieval details. The second line will represent the public key that the callee would like the voicemail to be encrypted with. This can be either the public key of the callee or the public key of a callee trusts. This mechanism is in place such that if the callee has a resource that is used for decrypting messages, such as a voicemail server, they can provide the details to the caller so that the callee's voicemail server can successfully retrieve and decrypt the voicemail. The third line will describe the type of key being provided including the length of the key. The fourth line of the message body will represent a length of time in seconds that the caller will have to respond with the voicemail retrieval details. As mentioned before, this can be used to control the length of voicemail that the caller will generate.

Once the voicemail notification request message has been received by the caller the caller can provide details to the callee of how to retrieve message later. It is at the discretion of the caller to record the message prior to sending back retrieval information to callee. If the caller records the message before providing details then the response time is crucial, as the callee has passed a time value which represents the amount of time it will wait for a response. If the caller provides the retrieval information prior to recording message, then it is free to encrypt message after SIP call flow has been completed. The information that the caller will have to provide to the callee is a unique message identification number such that the callee can correctly identify the message. The caller must also provide a URI that represents the location that will host the voicemail message. This can either be the caller or its proxy server used during the call. These are the only two eligible URI since we can have and can only verify the caller and its proxy prior to

voicemail retrieval. The caller must provide a length of time represented in seconds that will represent the amount of time that the callee will have to retrieve the message. It is up to local mechanisms on the caller's side to expire and delete unretrieved messages.

```
OK sip:alice@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
;received=192.0.2.222
Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
;received=192.0.2.201
Max-Forwards: 68
From: Bob <sip:bob@biloxi.example.com>;tag=314159
To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 OK
Content-Length: 250
Content-Type: text/encrypted
VOICEMAIL NOTIFICATION
alice@atlanta.example.com
40000000
```

Figure 5-3. Example DSIP voicemail notification phase voicemail notification message

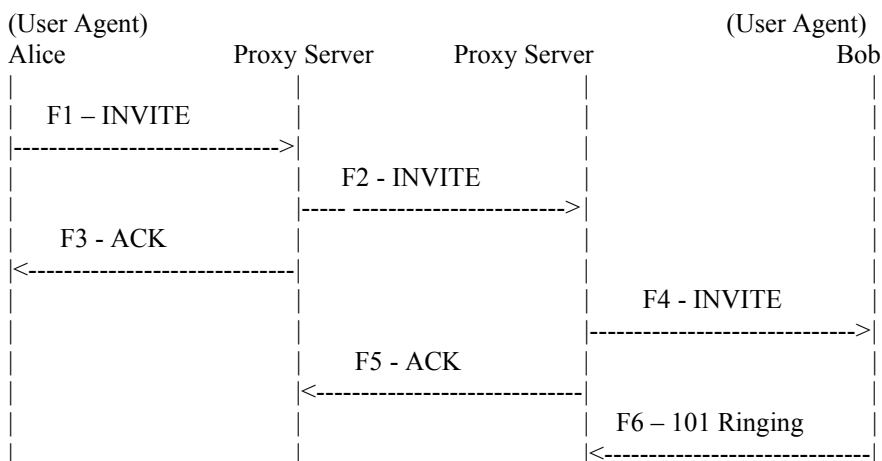
Points of interest include the “ContentLength” header, which describes the length of the message body and originally indicated that no message body content existed. For the voicemail notification request message, it will be necessary to modify this “ContentLength” header and insert the “ContentType” header to describe the message body. The first line of the message body will indicate “VOICEMAIL NOTIFICATION” to indicate to the callee that this is the voicemail retrieval information. The second line will represent the message identification number which is a 16-bit integer value. This is the number that will identify the caller's voicemail. The third value will represent a length of time in seconds that the callee will have to retrieve the voicemail before it expires and will no longer be available. The most important feature is that all this information is required to be encrypted using the callee's public key. The callee will decrypt this

information and use the first line as a sanity check to make sure the message was encrypted correctly. The message after decryption should have the first line read "VOICEMAIL NOTIFICATION".

## CHAPTER 6 VOICEMAIL RETRIEVAL PHASE

In this chapter we present how a callee can retrieve a voicemail message from a caller if the callee wishes to do so.

The voicemail retrieval phase is tasked with retrieving the remote voicemail message using the information provided by the original caller. In principle, it is identical to the function provided by the GTML (get mail) command in DMTP [DDK06]. The voicemail notification phase will provide the original callee retrieval details including the a unique message identification number such that the callee can correctly identify the message, a URI that represents the location that will host the voicemail message, and a length of time represented in seconds that will represent the amount of time that the callee will have to retrieve the message. There is no necessary for password exchange between the voicemail storage facility and a caller during the retrieval process because the entire voicemail is already encrypted with the authorized retriever's public key, with sufficient strength that it will not be worthwhile for others to attempt to crack.



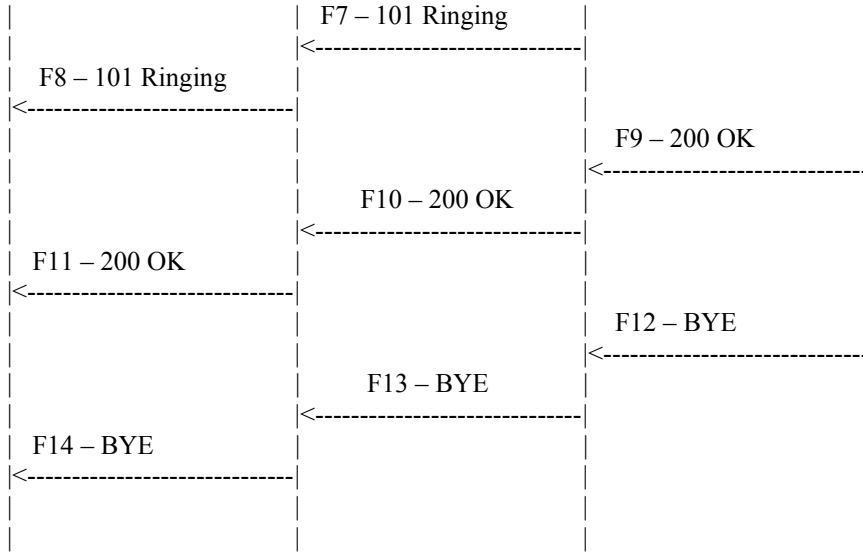


Figure 6-1. SIP Call flow of the voicemail retrieval phase

The first step of the voicemail retrieval phase is to modify the INVITE request. It is important to note that the voicemail notification phase indicates that a voicemail can be stored on any SIP part of the chain of trust built when the original call was accepted. It is up to the voicemail notification phase to perform validity checks and determine if the storage facility specific is in the acceptable chain of trust. The retrieval client will attempt to initiate a SIP conversation with the retrieval server by sending a modified INVITE request.

```

INVITE sip:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 151
VOICEMAIL RETRIEVAL
HKDJHKGH7686999335K3H45K3H45K34H5K3H5KJ87893H4T8DKWHFLDSFJSLVJ
  
```

Figure 6-2. Example DSIP voicemail retrieval phase voicemail retrieval request message

Points of interest include the “ContentLength” header, which describes the length of the message body. For the voicemail retrieval request message, it will be necessary to modify this “Content-Type” header and replace the content-body to only include voicemail retrieval information. The first line of the message body will indicate “VOICEMAIL RETRIEVAL” to indicate to the voicemail retrieval server that the caller desired to retrieve a voicemail. The second line will represent the unique message identification number that was left by the original caller such that the voicemail retrieval client can such correctly identify the message

After the voicemail server has received the information about which message the retrieval client is requesting, it will send back the content of the media file. It is important to not that the media being sent back can be any type of media including pictures, voice files, or any other media. The description about the incoming media will simply be noted in the “Content-Type” of the response message, but it will be assumed that the retrieval client understands that this information being returned is in need o being decrypting.

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1
;received=192.0.2.222
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1
;received=192.0.2.111
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
;received=192.0.2.101
Record-Route: <sip:ss2.biloxi.example.com;lr>,
<sip:ss1.atlanta.example.com;lr>
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.example.com>;tag=314159
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 2 INVITE
Content-Type: application/mpg
Content-Length: 151
Sdhf723jh2g3kj2hsd2z8UHWI@u2hk2jhkh2u3h89udfdkjhskghsdkjghkj2hkh
```

Figure 6-3. Example DSIP voicemail retrieval phase voicemail retrieval data message

The response message will simply be a normal sip response with the “Content-Type” modified to match the media being sent back with the “Content-Length” field modified to match the length of the body .This information is simply piggybacked on the final 200 OK response sent back to the caller.

## CHAPTER 7 SUMMARY

The explosion of the internet has generated a wide array of technological advances related to packet based networks. As a result it is becoming more reasonable for businesses and consumers to communicate through their internet connections as opposed to their traditional telephone carrier networks. VoIP will provide cost savings as they reduce their reliance on the Public Switched Telephone Network (PSTN) and require maintenance of only one network of intermixed voice and data. With the explosion of broadband subscription among consumers, both business and residential users are able to take advantage of VoIP technologies.

VoIP spam is a serious threat to the growth of VoIP. The future of VoIP is very bright, however the success of spam on email technology almost guarantees that there will be an attempt to abuse the cost effectiveness of VoIP communications. Spam over Internet Telephony (SPIT) not only could be a major annoyance in the future, it has the realistic potential to cripple the growth of VoIP, and thus the usefulness of the cost effective technology. If left uncontrolled SPIT will be more than annoying, SPIT will also have an enormous economic impact as it will drain the productivity of users.

This project proposed DSIP as an effective means to control VoIP spam. DSIP is a modification of the SIP specification and extends its feature set to include internal mechanisms to identify and mitigate the onset of SPIT. The design of DSIP is heavily influenced by the Differentiated Mail Transfer Protocol (DMTP) proposed by Duan, Dong, and Gopalan to control email spam [DDK06]. DSIP has many advantages with

respect to SPIT control. DSIP will help reduce SPIT from reaching recipients, thus preventing annoying and malicious activity and not disturbing productivity. DSIP gives end users greater flexibility over who may contact them and leave multimedia messages. And finally, DSIP imposes legitimate callers and recipients with little to no overhead in their normal SIP transactions and multimedia communications, while penalizing abusive callers with the expense of disk space and CPU cycles, in an effort to globally prevent the onset of SPIT and denial of service attacks. With low impact on the current SIP specification, DSIP modifies one of the fastest growing voice technologies to enable an effective means of controlling the onset of VoIP spam.

## APPENDIX A CLASSIFICATION ROUTING IMPLEMENTATION

Implementation of the DSIP's classification routing phase comprised of modification to a SIP proxy server. The classification routing phase is responsible for routing the incoming call based on the classification of the caller. The classification routing phase was implemented on the siproxd open source project ( <http://siproxd.sourceforge.net/> ). Siproxd heavily relies on the libosip2 library (<http://www.gnu.org/software/osip/doc/html/> ) and is written in C.

The SIP definition RFC gives clear recommendations for how SIP proxy servers should process each message they receive. The RFC indicates that these steps are merely a suggestion and not required for RFC compliance. Siproxd was able to clearly document and implement the RFC provided strategy in their code. The RFC defines that an incoming request message should be processed according to the request processing checklist (Table 8-1).

1. Validate the request (Section 16.3)
2. Preprocess routing information (Section 16.4)
3. Determine target(s) for the request (Section 16.5)
4. Forward the request to each target (Section 16.6)
5. Process all responses (Section 16.7)

Table A-1. RFC3261 Request Processing Checklist

It is during this first phase of validating the request that I implement the classification routing phase. The RFC further defines the request validation phase with the request validation checklist (Table 8-2).

1. Reasonable Syntax
2. URI scheme
3. Max-Forwards
4. (Optional) Loop Detection
5. Proxy-Require
6. Proxy-Authorization

Table A-2. RFC3261 Request Validation Checklist

If the request should fail any item in the checklist the proxy is instructed to act as a user agent and respond with an error message. My implementation simply appends the RFC3261 Request Validation Checklist to include a classification check (Table 8-3).

1. Reasonable Syntax
2. URI scheme
3. Max-Forwards
4. (Optional) Loop Detection
5. Proxy-Require
6. Proxy-Authorization
7. Classification Check

Table A-3. DSIP Request Validation Checklist

In this example implementation of the classification routing phase it will be assumed that a database of classified callers has already been established. Based on the type of a classification for a user, the request will be either processed and continue on the normal SIP request path or the SIP call flow will be interrupted. The classification routing implementation makes use of three data structures to maintain the list of users under each classification. The blacklist array holds URIs of callers that are currently blacklisted, the whitelist array holds URIs of callers that are currently whitelisted, and the verified array holds callers of users that are currently verified. If an incoming request's sender does not exist in either of these data structures, the user is deemed to be greylisted and the request is forwarded on to DSIP's human verification phase. There is not need for a fourth data structure to maintain greylisted users as this data structure could become

vulnerable to a DoS attack and is resourcefully wasteful. The human verification phase has an eventual end result which will insert the user into one of the three previously defined data structures.

The classification routing phase first determines if the message to be evaluated is an INVITE request. There is no need to monitor any other message outside of the required call setup messages, which all begin with the INVITE request. The sender of the request is simply searched in a linear fashion to determine if the user exist in the blacklist, whitelist, or verified array. There are only two possible scenarios as a user could exist in one data structure or in no data structure.

These scenarios are clearly defined and should proceed in a deterministic fashion. If the user exists in a single data structure then the user should be classified by that data structure. If the user does not exist in any of the data structures, then the user should assume the default classification, which is unclassified, and the user's request should be forwarded to the human classification phase. The software that end recipients use to interact with the data structures should perform some checks to ensure that a sender does not exist in multiple data structures.

### **Code Insertion**

#### **proxy.c**

```
extern int blacklist_array_counter;  
extern char *blacklist_array[];  
int blacklist_bool;  
int temp1;
```

```
extern int whitelist_array_counter;  
extern char *whitelist_array[];  
int whitelist_bool;
```

```
extern int verified_array_counter;  
extern char *verified_array[];
```

```

int verified_bool;

char temp_string[100];
strcpy(temp_string,request->from->url->username);
strcat(temp_string,"@");
strcat(temp_string,request->from->url->host);

if(MSG_IS_INVITE(request))
{
    printf("THE INCOMING MESSAGE IS AN INVITE REQUEST ADDRESSED
FROM: %s\n", temp_string);

    blacklist_bool = 0;
    temp1 = 0;
    for(temp1 = 0; temp1 < blacklist_array_counter; temp1++)
    {
        if(strcmp(blacklist_array[temp1],temp_string) == 0)
        {
            printf("MATCH FOUND IN BLACKLIST ARRAY -- %i\n",
strcmp(blacklist_array[temp1],temp_string));
            blacklist_bool = 1;
        }
    }

    whitelist_bool = 0;
    temp1 = 0;
    for(temp1 = 0; temp1 < whitelist_array_counter; temp1++)
    {
        if(strcmp(whitelist_array[temp1],temp_string) == 0)
        {
            printf("MATCH FOUND IN WHITELIST ARRAY -- %i\n",
strcmp(whitelist_array[temp1],temp_string));
            whitelist_bool = 1;
        }
    }

    verified_bool = 0;
    temp1 = 0;
    for(temp1 = 0; temp1 < verified_array_counter; temp1++)
    {
        if(strcmp(verified_array[temp1],temp_string) == 0)
        {

```

```

        printf("MATCH FOUND IN VERIFIED ARRAY -- %i\n",
strcmp(verified_array[temp1],temp_string));
        verified_bool = 1;
    }
}

//Classification Order of Priority
//The whitelisted classification supercedes all other classifications
//The greylisted classification only supercedes blacklisted classification
//The blacklisted classification is the lowest ranking classification
//If the caller does not exist in either of these data structures then the
//user is forwarded on to the human verification phase.

if(whitelist_bool)
{
    printf("The SIP call flow will proceed as normal.\n");
    printf("Caller is ALLOWED to leave voicemail directly on callee's
machine\n");
}
else if(verified_bool)
{
    printf("The SIP call flow will proceed as normal.\n");
    printf("Caller is NOT ALLOWED to leave voicemail directly on callee's
machine\n");
}
else if(blacklist_bool)
{
    printf("The SIP call flow will be interrupted!\n");
}
else
{
    printf("The Caller is not classified.");
    printf("The caller will be forwarded onto the Human Classification Phase\n");
}
}

```

### **siproxd.c**

```

int blacklist_array_counter = 2;
char *blacklist_array[] = { {"amadhosingh_black@stats.hcs.net"},
    {"amadho_black@amadho.com"}
};

int whitelist_array_counter = 2;

```

```

char *whitelist_array[] = { {"amadhosingh_white@stats.hcs.net"},
                             {"amadho_white@amadho.com"}
                           };

int verified_array_counter = 2;
char *verified_array[] = { {"amadhosingh_verified@stats.hcs.net"},
                             {"amadho_verified@amadho.com"}
                           };

```

## Routing Verification

### verified\_example.txt

```

INVITE sip:user2@amadho.hcs.net SIP/2.0
Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhds Max-Forwards: 70
To: amadhosingh_white <sip:amadhosingh_white@stats.hcs.net>
From: amadhosingh_verified
<sip:amadhosingh_verified@stats.hcs.net>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.server1.com
CSeq: 314159 INVITE
Contact: <sip:user1@pc33.server1.com>
Content-Type: application/sdp
Content-Length: 0

```

### blacklist\_example.txt

```

INVITE sip:user2@amadho.hcs.net SIP/2.0
Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhds Max-Forwards: 70
To: amadhosingh_white <sip:amadhosingh_white@stats.hcs.net>
From: amadhosingh_black <sip:amadhosingh_black@stats.hcs.net>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.server1.com
CSeq: 314159 INVITE
Contact: <sip:user1@pc33.server1.com>
Content-Type: application/sdp
Content-Length: 0

```

### greylist\_example.txt

```

INVITE sip:user2@amadho.hcs.net SIP/2.0
Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhds Max-Forwards: 70
To: amadhosingh_white <sip:amadhosingh_white@stats.hcs.net>
From: amadhosingh_grey <sip:amadhosingh_grey@stats.hcs.net>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.server1.com

```

CSeq: 314159 INVITE  
 Contact: <sip:user1@pc33.server1.com>  
 Content-Type: application/sdp  
 Content-Length: 0

### **whitelist\_example.txt**

INVITE sip:user2@amadho.hcs.net SIP/2.0  
 Via: SIP/2.0/UDP pc33.server1.com;branch=z9hG4bK776asdhs Max-Forwards: 70  
 To: amadhosingh\_white <sip:amadhosingh\_white@stats.hcs.net>  
 From: amadhosingh\_white <sip:amadhosingh\_white@stats.hcs.net>;tag=1928301774  
 Call-ID: a84b4c76e66710@pc33.server1.com  
 CSeq: 314159 INVITE  
 Contact: <sip:user1@pc33.server1.com>  
 Content-Type: application/sdp  
 Content-Length: 0

### **Whitelist Example**

```
# sipsak -f /home/amadhosingh/project_directory/sipsak/sip_files/whitelist_example.txt -s sip:amadho@amadho.hcs.net
THE INCOMING MESSAGE IS AN INVITE REQUEST ADDRESSED FROM:
amadhosingh_white@stats.hcs.net
MATCH FOUND IN WHITELIST ARRAY -- 0
The SIP call flow will proceed as normal.
Caller is ALLOWED to leave voicemail directly on callee's machine
```

### **Greylist Example**

```
# sipsak -f /home/amadhosingh/project_directory/sipsak/sip_files/greylist_example.txt -s sip:amadho@amadho.hcs.net
THE INCOMING MESSAGE IS AN INVITE REQUEST ADDRESSED FROM:
amadhosingh_grey@stats.hcs.net
The Caller is not classified.The caller will be forwarded onto the Human Classification Phase
#
```

### **Blacklist Example**

```
# sipsak -f /home/amadhosingh/project_directory/sipsak/sip_files/blacklist_example.txt -s sip:amadho@amadho.hcs.net
THE INCOMING MESSAGE IS AN INVITE REQUEST ADDRESSED FROM:
amadhosingh_black@stats.hcs.net
MATCH FOUND IN BLACKLIST ARRAY -- 0
The SIP call flow will be interrupted!.
```

**Verified Example**

```
sipsak -f /home/amadhosingh/project_directory/sipsak/sip_files/verified_example.txt -s
sip:amadho@amadho.hcs.net
THE INCOMING MESSAGE IS AN INVITE REQUEST ADDRESSED FROM:
amadhosingh_verified@stats.hcs.net
MATCH FOUND IN VERIFIED ARRAY -- 0
The SIP call flow will proceed as normal.
Caller is NOT ALLOWED to leave voicemail directly on callee's machine
#
```

## LIST OF REFERENCES

- [DDK06] Zhenhai Duan, Yingfei Dong, Kartik Gopalan, “DMTP: Controlling Spam Through Message Delivery Differentiation”, **To appear In Proc. Networking 2006**, Coimbra, Portugal, May 15-19, 2006.
- [RFC3261] Camarillo, Handley, Johnston, Peterson, Rosenberg, Schooler, Schulzrinne, Sparks. “SIP: Session Initiation Protocol.” RFC 3261. 2002. Accessed: April 10, 2006. <http://www.ietf.org/rfc/rfc3261.txt>
- [CISCO] Cisco Systems. "Understanding Voice over IP Protocols." 2002. Cisco Systems. Accessed: June 6, 2005. [www.cisco.com/application/pdf/en/us/guest/tech/tk587/c1506/ccmigration\\_09186a008012dd36.pdf](http://www.cisco.com/application/pdf/en/us/guest/tech/tk587/c1506/ccmigration_09186a008012dd36.pdf)
- [CJS06] Cunningham, Johnston, Summers. “Session Initiation Protocol (SIP) Basic Call Flow Examples.” 2003. Accessed: April 10, 2006. <http://ietfreport.isoc.org/rfc/PDF/rfc3665.pdf>
- [DK06] Dantu, Kolan. “Detecting Spam in VoIP Networks” USENIX: The Steps to Reducing Unwanted Traffic on the Internet Workshop. 2005. Pg. 31–37. Accessed: April 10, 2006. <http://www.usenix.org/events/sruti05/tech/dantu.html>
- [FKW06] Fries, Kuhn, Walsh. “Recommendations of the National Institute of Standards and Technology.” 2005. Pg. 39-46. Accessed: April 10, 2006. <http://csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf>
- [G06] Grace. “Number of VoIP Subscribers more than Triples in 2005 Reaching 4.2 Million; Expected to Grow to 18 Million by 2009.” 2006. Accessed: April 11, 2006. [http://www.tiaonline.org/business/media/press\\_releases/2006/PR06-19.cfm](http://www.tiaonline.org/business/media/press_releases/2006/PR06-19.cfm)
- [KSST05] Katz, Schwartz, Sterman, Tschofenig. “SPAM for Internet Telephony (SPIT) Prevention using the Security Assertion Markup Language (SAML).” 2005. Accessed: April 10, 2006. <http://www3.ietf.org/proceedings/05nov/IDs/draft-schwartz-sipping-spit-saml-00.txt>
- [M06] Mark. “VoIP Rings Up Explosive Growth.” 2006. Accessed: April 10, 2006. <http://www.internetnews.com/stats/article.php/3587946>
- [T06] Teitelbaum. “SIP Spam: the Coming Storm.” 2004. Accessed: April 10, 2006. <http://www.internet2.edu/sip.edu/200406-workshop/talks/20040616-teitelbaum-spam.pdf>

