Design for Elimination of Clipping Effects and Ghost Ringing in SIP initiated IPSec based VoIP

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By:
JASPREET SINGH
Roll No. 801032010

Under the supervision of:
Dr. V.P. SINGH
(Assistant Professor)

Mr. ASHISH AGGRAWAL
(Assistant Professor)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2012

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "Design for *Elimination of Clipping Effects & Ghost Ringing in SIP initiated IPSec based VoIP*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. V.P. SINGH and Mr. ASHISH AGGRAWAL* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(JASPREET SINGH)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. V.P. SINGH)
Assistant Professor,
CSED

(Mr. Ashish Aggrawal)
Assistant Professor,
CSED

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean
Academic Affairs
Thapar University
Patiala

ACKNOWLEDGEMENT

First of all I am thankful to God for blessings and showing me the right decision. With his mercy, it has been possible for me to reach so far.

I would like to express sincerest thanks to my thesis supervisors Dr. V.P. Singh and Mr. Ashish Aggrawal, for their inspiration, guidance, stimulating suggestions, immense help and support throughout the period of this research work. They have provided me all the necessary resources including motivation and research environment without which it would not have been possible to complete this work. It was a great opportunity for me to do this work under their supervision.

I am thankful to the authors whose work I have consulted and quoted in this work.

I lack words to express my cordial thanks to all my friends for their useful comments and constructive suggestions during all the phases of my life.

Finally, I convey deep sense of gratitude towards my family members for their moral and financial support and encouragement without which it would not have been possible to bring out this thesis.

ABSTRACT

Security is an important parameter in Voice over Internet Protocol (VoIP) which faces the inherited loopholes of IP networks and few of its own. Security in VoIP has two main aspects. First, securing the call signaling, i.e. securing the IP traffic used for the establishment of the call using Session Initiation Protocol (SIP), which is provided by Secure Multimedia Internet Mail Extension (S/MIME) or Transport layer Security (TLS). Second, securing the call itself, i.e. securing the Real Time transport Protocol (RTP) media session over the User Datagram Protocol. Security for RTP Media is provided at two different layers. Secure Real Time transport Protocol (SRTP) is a security profile of RTP which provide confidentiality service at application layer and Internet Protocol Security (IPSec) at the network layer. IPSec, when initiated by utilizing Offer Answer model of SIP, i.e. SIP initiate IPSec, uses less number of roundtrips with the help of Multimedia Internet Keying (MIKEY) protocol.

Call establishment in SIP initiated IPSec deals with cryptographic processing at the caller and the recipient end. Parameter negotiation for IPSec is done through the SIP INVITE message in calling phase and 200OK response in the answering phase when the responder picks up the phone. This IPSec cryptographic processing at caller end makes it unable to send and receive the RTP media packets at the beginning of the RTP session. This situation is called as call clipping. The caller faces media transmit clipping and media reception clipping at the start of the call due to which the caller can neither send nor receive the RTP media packets for first few milliseconds. Ghost ringing is the byproduct of the solution of clipping effect which is also dealt with the use of another provisional response at the cost of ringing delay. An Algorithm for eliminating clipping effect (Caller's transmission clipping, Caller's Reception Clipping) is described. The SIP's extension of PRACK has been used as an essential aid in this solution to provide reliability to unreliable response messages that are used for transporting IPSec cryptographic parameters. The Algorithm describes the process for shifting the IPSec cryptographic processing from the answering phase to the calling phase and a way to eliminate ghost ringing.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Internet Protocol (IP) Telephony is the transmission of voice, fax and related services over packet-switched IP-based networks. Voice over IP (VoIP) has raised much interest as the number of providers offering VoIP services has increased. IP Telephony, rather than PSTN (Public Switched Telephone Network) which is circuit switched technology, is based on packet switching. The voice transmission, instead of using a dedicated line, is broken in numerous packets. These packets look for the most efficient route individually, and later are reorganized at the IP address where the voice was originally sent. This technology allows numerous users to send information over the same line, providing a more efficient utilization of the telecommunications infrastructure.

1.1 VoIP Standards & Protocol Infrastructure

The VoIP infrastructure consists of endpoints i.e. Phones/Softphones, control nodes, gateway nodes and the IP-based network. The VoIP data processing consists of the following three steps [1]:

1- Signaling
2- Encoding and Transport
3- Gateway control

***Signaling*:**

ITU (International Telecommunication Union) has proposed a large, comprehensive set of protocols under the general umbrella of H.323. IETF (Internet Engineering Task Force) has proposed the Session Initiation Protocol (SIP)**.** Both of these protocols are used for Signaling where SIP is primarily used due to extensive support to IP oriented applications. H.323 and SIP, both are application layer protocols used for the control and signaling of the call. H.323 is a suite of protocol in which different protocols are defined to provide different services as shown in Figure 1.

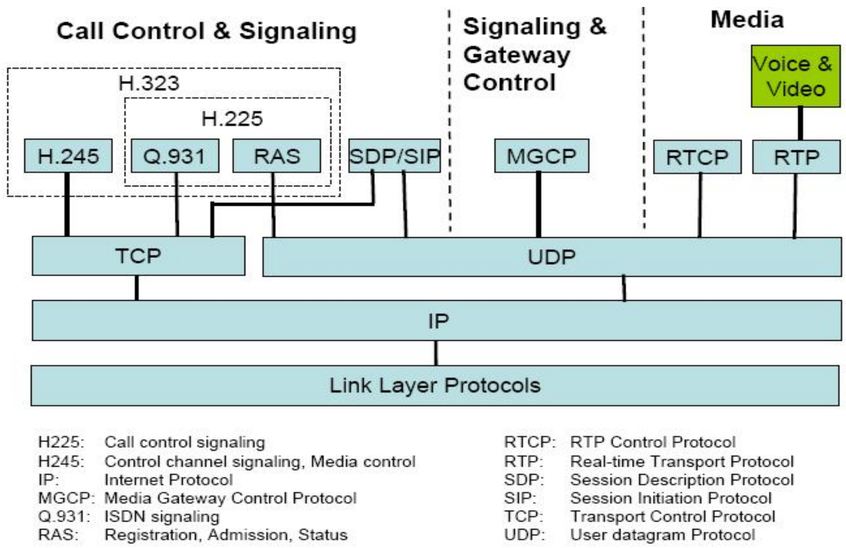

Figure 1: VOIP Protocol Stack [2]

Encoding & Transport:

There are various encoding standards providing different Codecs(Coder-Decoders) proposed by ITU, for example G.711 (PCM-Pulse Code Modulation) provides 64Kbps bit rate. Transport of media packets as shown in Figure 1 is done by RTP (Real Time Protocol) and RTCP (RTP Control Protocol). RTCP is a Control protocol which controls the transmission of media RTP packets and provides a mechanism for QOS. RTP provides all the major facilities of TCP such as sequencing, time-stamping and some other management services as VoIP uses UDP as transport protocol which doesn't provide any reliable delivery service.

Gateway Control:

Gateway control is done by MGCP (Media Gateway Control Protocol). MGCP is a protocol established by the collaboration of ITU and IETF (Internet Engineering Task Force) with compromised features of both the telecommunication and TCP/IP world as shown in Figure 1. MGCP is used to provide a way to interconnect a large number of IP telephony gateways, allowing them to work together. MGCP provides facilities to Call Agent (CA) that performs the intelligence of all call control operations. Media Gateway Controller (MGC) carries out all media processing and conversion as directed by MGCP.

1.2 Securing VOIP

Security for Voice over IP (VoIP) is a critical issue of IP Security. VoIP is very sensitive to delays. When security in VoIP is employed, the User Agents such as Softphones takes more time in the cryptographic processing. Security for VoIP can be achieved in different ways and can be divided into two main aspects. First, securing the *Call Signaling* i.e. the IP traffic used for establishing the call and second, securing the call itself, here referred to as the *Media Session*. Both are briefly stated below:

1.2.1 Secure VoIP Call Signaling

VOIP signaling is done for initiating and disconnecting a call. It contains all the parameters of VOIP media session that it is going to establish. Hence it is important to secure VOIP signaling. The VoIP signaling is secured using S/MIME (Secure Multipurpose Internet Mail Extension) profile of MIME (Multipurpose Internet Mail Extension). It can also be secured using MIKEY (Multimedia Internet Keying) for end-to-end authentication and keying and TLS (Transport Layer Security) for hop-by-hop protection of SIP (Session Initiation Protocol) messages [2].

1.2.2 Secure VoIP Media Session

VoIP media sessions are secured at different layers but the most prominent and flexible way is at application layer. Although, confidentiality of this session is provided at network layer by Internet Protocol Security (IPSec). For securing VoIP media session, the prime requirement is to deploy the security protocol which uses agile User Datagram Protocol (UDP) at transport layer [3]. At application layer, security is achieved by using SRTP (Secure Real Time Transport Protocol). SIP initiated Tunnels at network layer are used for IPSEC whose key negotiation is done by MIKEY using SIP-SDP message structure. SDP is a Session Description Protocol which provides a formal structure of description for Media session i.e. the codec profile to use, session id, session name, ports and IP addresses etc.

1.3 Thesis Organization

The thesis outline is as follows. Chapter 1 contains introduction to VoIP technology and its security. Chapter 2 is composed of the technologies involved in this thesis. SIP, SDP, IPSec, MIKEY, MIME and S/MIME are described in this chapter. Chapter 3 contains the literature survey on the existing infrastructure of SIP initiated IPSec using MIKEY keying protocol. Chapter 4 explains the problem statement of delays in the call establishment when SIP initiated IPSec is used with MIKEY keying protocol. Chapter 5 contains the solution design and Algorithm. It explains the use of provisional messages for MIKEY reply for eliminating cryptographic delays at time sensitive phase. Chapter 6 contains the conclusion and the future scope.

CHAPTER 2

TECHNOLOGIES INVOLVED

There are various protocols involved in establishing, maintaining and disconnection of the VOIP session. The most important protocols and technologies are explained in this chapter. This chapter provides an overview of the technologies that are used in the State of art in the next chapter (chapter 3). SIP is explained in a more extensive way as it is a key protocol for obtaining the objective of this thesis.

2.1 SIP (Session Initiation Protocol)

To establish a VoIP session between two persons/hosts, signaling is needed to find each party of the call. SIP as explained in [5] is such a signaling protocol.

2.1.1 General

When trying to make a VoIP call, the caller needs to find the callee. In the world of the *circuit switched telephony*, each subscriber has a unique telephone number identifying the telephone of the subscriber. The relationship between the subscriber e.g. Mr. XYZ and his telephone number is static and usually found in the phone book. When a call is made a connection is established between the well known locations of the caller and the callee.

IP telephony uses IP addresses to find the path from the caller to the callee. The connection between Mr. XYZ and his IP address is loose and may change over time. For instance the IP telephony client might have got its IP address dynamically from DHCP (Dynamic Host Configuration Protocol) server and may differ each time the client connects. If the caller do not know the IP address of the callee, SIP is a way to find out and establish the connection. If the caller already knows the IP address of the callee the session parameters could be negotiated directly by other means than SIP and SIP can of course still be used as a convenient and consistent way of establishing the connection. SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. SIP transparently supports name mapping and redirection services, which supports personal mobility [6]**.** Users can maintain a single externally visible identifier regardless of their network location.

SIP supports five facets of establishing and terminating multimedia communications:

User location: Determination of the IP address of the end system to be used for communication.

User availability: Determination of the willingness of the called party to engage in communication.

User capabilities: Determination of the media and media parameters to be used.

Session setup: Ringing, establishment of session parameters at both called and calling party.

Session management: Transfer and termination of sessions, modifying session parameters, and invoking services.

SIP can be used with other IETF protocols to build a complete multimedia architecture. Typically, these architectures will include protocols such as the Real-time Transport Protocol (RTP) for transporting real-time data, Media Gateway Control Protocol (MEGACO) explained in [7]for controlling gateways to the Public Switched Telephone Network (PSTN) and the Session Description Protocol (SDP) for describing multimedia sessions. SIP is used in conjunction with other protocols in order to provide complete services to the users.

The nature of the services provided make security particularly important. Hence SIP provides a suite of security services, which include denial-of-service prevention, authentication (both user to user and proxy to user), integrity protection and encryption like privacy services.

2.1.2 SIP architecture

There are five entities in SIP named as Registrar, Location Server, Proxy, UA (User Agent) and Redirect server.

Entities called Clients interacting in SIP scenarios are known as User Agents (UAs). User Agents may operate in two fashions -

1. User Agent Client (UAC): It generates requests and sends those to the servers.
2. User Agent Server (UAS): It gets requests, processes those requests and generates responses.

Entities called Servers are general part of the networks. They possess a predefined set of rules to handle the requests sent by clients. Servers can be of several types -

- *Proxy Server*: These are the most common type of server in a SIP environment. When a request is generated, the exact address of the recipient is not known in advance. So the client sends the request to a proxy server. The server on behalf of the client (as if giving a proxy for it) forwards the request to another proxy server or the recipient itself.
- *Redirect Server*: A redirect server redirects the request back to the client indicating that the client needs to try a different route to get to the recipient. It generally happens when a recipient has moved from its original position either temporarily or permanently.
- *Registrar*: One of the prime jobs of the servers is to detect the location of a user in a network. Users from time to time refresh their locations by registering (sending a special type of message REGISTER) to a Registrar server.
- *Location Server*: The addresses registered to a Registrar are stored in a Location Server.

The Registrar receives registrations and requests updates of the *Location Server*, which keep track of the UA's. Each Registrar belongs to a domain in a similar way as a mail server can belong to a domain. The Proxy Server routes SIP messages on behalf of the

UA. *Redirect Server* directs UA's to alternate URI. Caller calls Callee using his SIP identity, a type of Uniform Resource Identifier (URI) [8] called a SIP URI. It has a similar form to an email address, typically containing a username and a host name. Example, sip:UA1@domain1.com, where domin1.com is the domain of UA1's SIP service provider.

Usually the Registrar, Location Server and the Proxy Server run on the same server. The UA registers with its registrar when it goes on line to tell the registrar that it is available. This is done with the SIP message REGISTER, as shown in Figure 2.1:

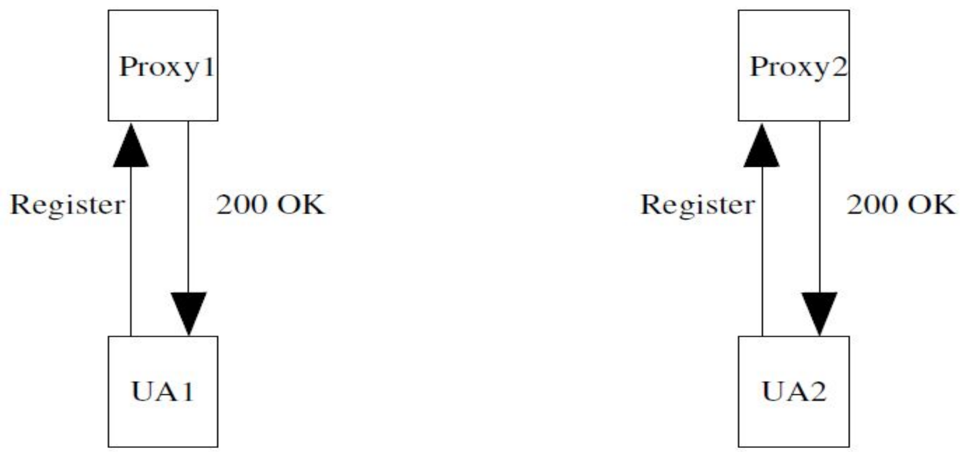

Figure 2.1: SIP Registration by User Agents at their respective Proxy/Registrar/Location Server [2]

An example of a REGISTER message of UA1 is shown below:

REGISTER sip:registrar.domain1.org SIP/2.0
To: <sip:ua1@domain1.org>
From: <sip:ua1@doedomain1.org>; tag=random_unique#
CallID: random#@domain1.org
Cseq: 4711 REGISTER
Contact: <sip:ua1@ua1pc.domain1.org:5060>;expires=1000
MaxForwards: 70
Via: SIP/2.0/UDP ua1pc.domain1.org:5060; branch=z9hG4bKrandomunique#
expires: 7200
ContentLength: 0

The REGISTER message contains information on how the UA can be reached, thus when the UA is registered in the registrar the UA can be reach by other UA's.

2.1.3 Making a call

A standard call can be described with the SIP trapezoid when UA1, the caller, wants to establish a call to UA2, the recipient, in which the following SIP messages are involved:

- INVITE: Used by the caller to initiate/invite for a call to the callee.
- Trying: A response from the next hop server that the INVITE message is received and processed. It is used by many UA but not mandatory.
- Ringing: An unreliable response message, alerting the caller that the callee has received the INVITE.
- 200 OK: A reliable response message, confirming the caller that the request has succeeded e.g. Call answered or Hang up.
- ACK (Acknowledgement): Establish the media session by acknowledging the receipt of 200 OK. It ensures the reliability of response messages.
- BYE: A message sent conveying a Hang up call or intentional disconnection of the call.

Example of INVITE message from caller UA1 to callee UA2 is shown below:

```
-----------------------------------------------------------------------------------------------
INVITE sip: UA2@domain2.org SIP/2.0
To: <sip:UA2@domain2.org>
From: <sip:UA1@domain1.org>; tag=random_unique#
CallID: unique#@domain1.org
Cseq: 4711 INVITE
Contact: <sip:UA1@UA1pc.domain1.org:5060;user=phone;transport=UDP>
MaxForwards: 70
Via: SIP/2.0/UDP UA1pc.domain1.org:5060;branch=z9hG4bKrandom_unique#
ContentType: application/sdp
ContentLength: 176
-- SDP body not shown --
-----------------------------------------------------------------------------------------------
```

The preferred and supported session parameters from the caller are encapsulated in the INVITE message in a SDP body, and the parameters chosen by the callee are encapsulated in the 200 OK message. The message ACK, BYE, 200 OK may go via the proxy. Figure 2.2 shows the standard SIP call setup between UA1 and UA 2 through their proxy servers. This call flow diagram shows the general call establishment between the users, the media session and the call termination. The DNS lookups are not shown.

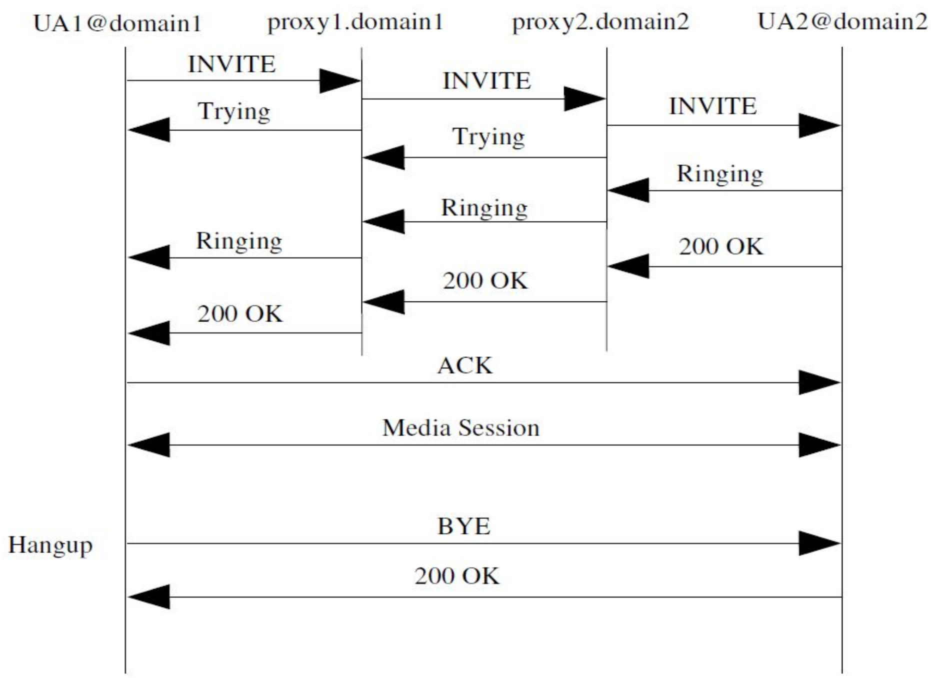

Figure 2.2: A standard SIP call setup showing Trapezoidal relationship between UA1 & UA2 via Proxies [9]

The fields in the SIP body are as follows:

To: logical Recipient of a request.

From: logical identity of the Initiator of the request.

CallID: A unique identifier to group a series of messages. It must be the same for all requests and responses sent by either UA in a dialog.

Cseq: It provides identification and order of transactions.

*Contac*t: It contains the the URI at which the UA would like to receive requests.

MaxForwards: It limits the number of hops a request can transit.

Via: It indicates the transport used for the transaction and identifies the location where the response is to be sent.

ContentType: It indicates the media type of the message body sent to the recipient (SDP Body not shown here, explained in next subsection).

ContentLength: The size of message body (SDP body size).

Example of a 200 OK response message from callee UA2 to caller UA1 is shown below:

```
----------------------------------------------------------------------------------------------------
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy1.domain2.org; branch=value; received=IP_address
Via: SIP/2.0/UDP proxy2.domain1.org; branch= value; received=IP_address
Via: SIP/2.0/UDP UA1pc.domain1.org; branch= value; received=IP_address
To: <sip:UA2@domain2.org> ; tag=random_unique#
From: <sip:UA1@domain1.org>; tag=random_unique#
CallID: unique#@domain1.org
Cseq: 4711 INVITE
Contact: <sip:UA2@UA2pc.domain2.org:5060;user=phone;transport=UDP>
MaxForwards: 70
ContentType: application/sdp
ContentLength: 176
-- SDP body not shown –
----------------------------------------------------------------------------------------------------
```

The header fields in this 200 OK Response are similar to those in a INVITE Request but with the following differences

Via: There are more than one via field. This is because each element through which the INVITE request has passed has added its identity in the Via field. Three Via fields are added by phone of UA1, the Proxy Server1 and Proxy Server2. The response retraces the path of INVITE using the Via fields. On its way back, each element removes the corresponding Via field before forwarding it back to the caller.

To: Note that the To field now contains a tag. This tag is used to represent the callee in a dialog.

Contact: It contains the exact address of UA2. So, UA1 doesn't need to use the proxy servers to find UA2 in the future.

It is a 2xx (200) response. However responses can be different depending on particular situations. The first digit of a Status-Code defines the category of response. SIP/2.0 allows six types of response. They are similar to those of HTTP.

- 1xx: Provisional: request received, continuing to process the request.
- 2xx: Success: the action was successfully received, understood, and accepted.
- 3xx: Redirection: further action needs to be taken in order to complete the request.
- 4xx & 5xx: Client & Server Errors respectively.
- 6xx: Global Failure: the request cannot be fulfilled at any server.

2.1.4 Reliable Provisional Responses (PRACK)

The Session Initiation Protocol (SIP) is a request-response technology for initiating and managing communication sessions. SIP defines two types of responses, provisional and final.

- **Final responses (2xx)** like 200 OK, conveys the result of the request processed and are sent reliably. Reliability here refereed to the reliable delivery of the response i.e using an explicit acknowledgement for the response received, which is must in Final responses. This reliability condition is fulfilled by sending back an ACK messages with respect to the Final 2xx responses.
- **Provisional responses (1xx)** like 183 Session in Progress, provides information on the progress of the request but are not sent reliably in SIP. But it was later observed that reliability is important in some cases, including some interoperability scenarios with the PSTN and the reliable mechanism is created [10].

The reliability mechanism works by mirroring the current reliability mechanisms for 2xx final responses to INVITE. Those requests are transmitted periodically by the Transaction User (TU) until a separate transaction, ACK, is received that indicates reception of the 2xx by the UAC. The reliability for the 2xx responses to INVITE and ACK messages are end-to-end. In order to achieve reliability for provisional responses, RFC 3262 describes nearly the same thing. Reliable provisional responses are retransmitted by the TU with an exponential back-off. Those retransmissions cease when a PRACK message is received.

The PRACK request plays the same role as ACK, but for provisional responses. Each provisional response is given a sequence number, carried in the RSeq header field in the response. The PRACK messages contain an RAck header field, which indicates the sequence number of the provisional response that is being acknowledged.

User Agent Server (UAS) Behavior

A UAS is allowed to send any non-100 provisional response to INVITE reliably. It is done when the initial INVITE request (the request whose provisional response is being sent reliably) contain a supported header field with the option tag 100rel. The UAS sends a non-100 provisional response reliably if the initial request contain a Require header field with the option tag 100rel. If the UAS is unwilling to do so, it than rejects the initial request with a 420 (Bad Extension) and include an Unsupported header field containing the option tag 100rel where the word rel stands for reliable.

A UAS is not allowed to send 100 Trying response reliably. Only provisional responses numbered 101 to 199 are sent reliably. If the request didn’t include either a Supported or Require header field indicating this feature, the UAS doesn’t send the provisional response reliably. 100 Trying response is a hop-by-hop message only. For this reason, the reliability mechanism described here, which is end-to-end, is not used.

User Agent Client (UAC) Behavior

When a UAC creates a new request, it is allowed to add reliable delivery of provisional responses for that request. To do so, it inserts a Require header field with the option tag 100rel into the request. If the UAC does not wish to insist on usage of reliable provisional responses, but merely indicate that it supports them if the UAS needs to send one, a Supported header is still included in the request with the option tag 100rel. The UAC include this in all INVITE requests. If a provisional response is received for an initial request, and that response contains a Require header field containing the option tag 100rel, the response is sent reliably. If the response is a 100 Trying (as opposed to 101 to 199), this option tag is ignored. Assuming the response is to be transmitted reliably, the UAC must create a new request with method PRACK. This request is sent within the dialog associated with the provisional response.

2.2 SDP (Session Description Protocol)

2.2.1 General

SDP is defined in [11].To establishes multimedia session over the Internet one needs to have a way of describing the attributes of the session. SDP is such a protocol. It is used to describe multimedia sessions in a format understood by the participants over a network. Depending on this description, a party decides whether to join a conference, when or how to join a conference. The caller advertises its capabilities over the network by sending SDP messages in SIP which contains description of the session like the name of the owner, the name of the session, the codec, the timing etc. Depending on this information the recipients/callee take a decision about participation in the session if it supports the given capabilities in the description.

If a client wants to establish a multimedia session, it can be announced through SDP, which may include [12]:

- Session name and purpose
- Time the session is active
- The type of media (video, audio)
- The transport protocol (RTP/UDP/IP, H.320)
- The format of the media (H.261 video, MPEG video)
- Information about where to receive those media (addresses, ports)

2.2.2 SDP used by SIP

To establish a VoIP call, the caller needs to negotiate session parameters with the callee. A call can consist of several multimedia channels e.g. voice and video etc., where each channel needs a unique set of parameters that describes the session. This negotiation is done with SIP. The caller sends a SDP body that is encapsulated in the SIP INVITE message, with proposed parameters. The callee responds with a SDP body in the OK message with the chosen parameters, so in just one round trip a common pair is negotiated.

Example of an INVITE message, with SDP body, containing negotiation parameters for media session from caller UA1 to callee UA2 is shown below:

```
--------------------------------------------------------------------------------------------------
INVITE sip: UA2@domain2.org SIP/2.0
To: <sip:UA2@domain2.org>
From: <sip:UA1@domain1.org>; tag=random_unique#
CallID: unique#@domain1.org
Cseq: 4711 INVITE
Contact: <sip:UA1@UA1pc.domain1.org:5060;user=phone;transport=UDP>
MaxForwards: 70
Via: SIP/2.0/UDP UA1pc.domain1.org:5060;branch=z9hG4bKrandom_unique#
ContentType: application/sdp
ContentLength: 176

v=0
o= 123 123 IN IP4 UA1pc.domain1.org
s=VOIP session
c=IN IP4 UA1pc.domain1.org
t=0 0
a=sendrecv
m=audio 32869 RTP/AVP 0
a=rtpmap:0 PCMU/8000/1
m=video 51372 RTP/AVP 31
--------------------------------------------------------------------------------------------------
```

The fields in the SDP body are as follows:

v = (protocol version) but also marks the beginning of the session description

o= (owner/creator and session identifier)

s= (session name)

c= (connection information not required if included in all media)

t= (time the session is active)

m= (media name and transport address) but also marks the beginning of the media description

a= (media attribute lines)

k= encryption key

b= bandwidth information

Example of a 200 OK response message including SDP's chosen parameters for media session, sent from callee UA2 to caller UA1 as shown below:

```
-----------------------------------------------------------------------------------------------------
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy1.domain2.org; branch=value; received=IP_address
Via: SIP/2.0/UDP proxy2.domain1.org; branch= value; received=IP_address
Via: SIP/2.0/UDP UA1pc.domain1.org; branch= value; received=IP_address
To: <sip:UA2@domain2.org> ; tag=random_unique#
From: <sip:UA1@domain1.org>; tag=random_unique#
CallID: unique#@domain1.org
Cseq: 4711 INVITE
Contact: <sip:UA2@UA2pc.domain2.org:5060;user=phone;transport=UDP>
MaxForwards: 70
ContentType: application/sdp
ContentLength: 176

v=0
o= 123 123 IN IP4 UA1pc.domain1.org
s=VOIP session
c=IN IP4 UA1pc.domain1.org
t=0 0
a=sendrecv
m=audio 32869 RTP/AVP 0
a=rtpmap:0 PCMU/8000/1
m=video 0 RTP/AVP 31
-----------------------------------------------------------------------------------------------------
```

Port number in media video is equals to 0(zero) which means that the callee doesn't want to or doesn't support video of this format. If a port number other than zero is present, it signifies a selected port or supported format.

2.3 IPSEC (Internet Protocol Security)

2.3.1 General

IPSEC [13] was designed to add security at the network layer i.e. adding security to all protocols above the network layer. IPSEC may make use of three protocols given below where the first two are for data protection and last one for Key negotiation:

- ESP: Encapsulating Security Payload Encrypts and/or authenticates data.
- AH: Authentication Header provides a packet authentication service.

- IKE: Internet Key Exchange negotiates connection parameters, including keys, for the other two i.e ESP and AH.

IPSEC as a term is a bit indistinct as sometimes it includes all three protocols band sometimes only a subset. E.g., if there is a manual key exchange there is no need for IKE and if the authentication that ESP provides is sufficient there is no need for AH. IPSEC is intended to protect traffic between hosts and is applied at the network layer. It does not supply the same kind of end to end security as protocols working at higher levels.

Higher level security protocols provide application to application security but IPSEC provides host to host security. Application to application is when an application, in this case a softphone, does all encryption and decryption and do not rely on any other application for that service. In the case of IPSEC the kernel of the Operating system handles all the security.

Which traffic to protect with IPSEC and which ciphers to use to encrypt and decrypt that information along with authentication is decided by the IPSec policy and IPSec Security Associations (SA) as described in next two sections.

2.3.2 SA (Security Associations)

Each IPSec secured connection is defined by a Security Associations (SA). For the peers to be able to encapsulate and decapsulate the IPSec packets, they need a way to store the secret keys, algorithms and IP addresses involved in the communication. All these parameters needed for the protection of the IP datagrams are stored in a security association (SA). The security associations are in turn stored in a security association database (SAD).

Each security association (SA) defines the following parameters [13]:

- *Source and destination IP* address of the resulting IPSec header. These are the IP addresses of the IPSec peers protecting the packets.
- *IPSec protocol (AH or ESP).*

- The *algorithm and secret key* used by the IPSec protocol.
- *Security Parameter Index (SPI*) which is a 32 bit number which identifies the security association.

Some implementations of the security association database allow further parameters to be stored:

- *IPSec mode* (tunnel or transport described in 2.3.5)
- *Size of the sliding window* to protect against replay attacks.
- *Lifetime* of the security association.

Example of a Security Association (SA) with manual keys configured in a linux configuration file of setkey utility on 192.168.1.100 computer in Transport mode [14]:

```
-------------------------------------------------------------------------------------------------
# ESP SAs doing encryption using 192 bit long keys (168 + 24
 parity and authentication using 128 bit long keys)

add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831
-A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df
-A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;
-------------------------------------------------------------------------------------------------
```

In the example there are two Security Associations, one for outgoing traffic and second for incoming traffic.

The first SA states that traffic from host 192.168.1.100 to 192.168.2.100 i.e outgoing traffic should be encrypted with 3des-cbc using the Encryption Key:

`0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831`

and authenticated with hmac-md5 using the Authentication Key:

`0xc0291ff014dccdd03874d9e8e4cdf3e6`

The second SA states that traffic from host 192.168.2.100 to 192.168.1.100 i.e incoming traffic should be decrypted with 3des-cbc using the Encryption Key:

`0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df` and authenticated with hmac-md5 using the Authentication Key:
`0x96358c90783bbfa3d7b196ceabe0536b`

2.3.3 IPSEC policy

IPSEC requires a Security Policy Database containing the IPSEC policies specifying which type of action to take for a specific packet. E.g. drop the packet, protect the packet or send in clear text. Decisions can be made on different fields in the packet such as source IP address, destination IP address, UDP or TCP transport protocol. The IPSEC Security Policy is actually a filter that decides which traffic to protect.

Example of a Security Policy (SP) [14] configured in a linux configuration file of setkey utility on 192.168.1.100 computer is shown below:

```
# Security policies
spdadd 192.168.1.100 192.168.2.100 any -P out ipsec
esp/transport//require
ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P in ipsec
esp/transport//require
ah/transport//require;
```

The above policy states that traffic from host 192.168.1.100 network with any source port to host 192.168.2.100 with any destination port must be protected with IPSEC and the policy to use (-P). The policy specifies the direction (in/out), the action to apply (ipsec/discard/none), the protocol (ah/esp/ipcomp), the mode (transport) and the level (use/require).

2.3.4 IKE (Internet Key Exchange)

Internet Key Exchange (IKE) is described in [15]. The IKE protocol is used for setting up IPSEC (ESP/AH) connections between two hosts/gateways. IKE negotiation has two phases. Phase 1 is used for proving each other's identity and set up a secure connection (ISAKMP-Internet Security Association Key Management Protocol’s SA or IKE SA) for phase 2. Phase 2 is used for negotiating IPSEC SA (ESP/AH) so that the secure data connection can be established. Since a SA is unidirectional they are negotiated in pairs to handle two-way traffic. The IKE SA can be used to negotiate more than one IPSEC SA. If there exist a IPSEC security policy that states that the traffic should be protected but there is no matching SA, IKE will try to negotiate a SA according to IKE configuration. If the negotiation fails and no new SA is created the traffic will be dropped. If the negotiation succeeds a new SA will be created.

Phase 1 can be either of two modes:

---

1. Main mode

Parameter negotiation

message 1: UA1 → crypto suites supported →UA2

message 2: UA1 ← chosen crypto suites ←UA2

DiffieHellman exchange

message 3: UA1 → g^a mod p →UA2

message 4: UA1 ← g^b mod p ←UA2

Send IDs and authenticate (Encrypted)

message 5: UA1 → g^ab mod p {"UA1", proof I'm UA1} → UA2

message 6: UA1 ← g^ab mod p {"UA2", proof I'm UA1} ← UA2

---

2. Aggressive mode

---

message 1: UA1 →g^a mod p, "UA1", crypto proposal → UA2

message 2: UA1 ←g^b mod p, "crypto choice, proof I'm UA2 ← UA2

message 3: UA1 → proof I'm UA1 →UA2

The main differences between mode 1 and 2 are, except for the number of packets, that mode 1 can negotiate the Diffie-Hellman (DH) number since the DH exchange is done in message 3&4 and that in aggressive mode sends Alice and Bob's identities unprotected.

Phase 2 (also known as "Quick mode")

Phase 2 IKE is a 3 message protocol that negotiates parameters for the phase 2 SA (ESP/AH SA), including cryptographic parameters and the SPI to identify the SA.

----
message 1: UA1→ X,Y, Kenc {CP, SPIUA1, nonceUA1, [g^a mod p]} → UA2

message 2: UA1← X,Y, Kenc {CPA, SPIUA2, nonceUA2, [g^b mod p]} ← UA2

message 3: UA1→ X,Y, ack → UA2

----

where: ● X, is the cookie pair generated in phase 1

● Y, 23bit number chosen by the phase 2 initiator to distinguish this phase 2 session from others within the same phase 1 session.

● CP, Crypto Proposal for SA

● CPA, Crypto Proposal Accepted

● [g^x mod p] optional DH values.

2.3.5 IPSEC modes

IPSEC can be used in either of two modes, Tunnel mode or Transport mode. In transport mode, the IPSEC information (AH and/or ESP) is inserted between the IP header and the rest of the packet. Transport mode, as shown in Figure 2.3, is used directly between two hosts to provide confidentiality and authentication.

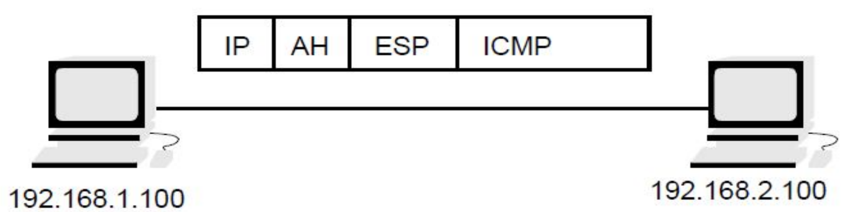

Figure 2.3: Two machines in Transport mode using AH and ESP [14]

In tunnel mode, the original packet is intact and a new IP header and IPSEC information is added outside as shown in Figure 2.4. This mode is used directly between two Proxies / Gateways to provide confidentiality and authentication i.e. creating a VPN.

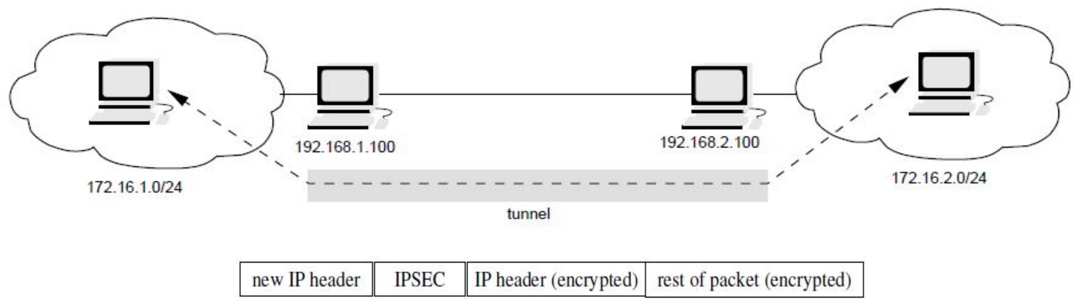

Figure 2.4: The two peers protecting the traffic through a Tunnel between two networks [14]

2.3.6 ESP (Encapsulated Security Payload)

ESP [16] can be used for encryption and integrity protection. ESP always uses encryption but integrity is optional as shown in Figure 2.5.The ESP protocol can both ensure the integrity of the packet using a HMAC and the confidentiality using encryption. After encrypting the packet and calculating the HMAC, the ESP header is generated and added to the packet. ESP is a rather odd header since it really encapsulates the encrypted data i.e. adding information both in front of the encrypted data and after. ESP can be used as integrity protection only if the special "null encryption" algorithm is used. Security Associations use ESPs for describing the encryption parameters like the algorithm and its mode.

A brief description of ESP Header fields is as follows:

- *SPI* - Security parameter index, identifies SA [4 octets]
- *Sequence number* - Used for protection against replay attacks and this has nothing to do with TCP sequence number [4 octets]
- *IV* - Initialization Vector is used by some cryptographic algorithms. The length of this field depends on the algorithm and once the SA is established the field length is fixed [variable]
- *Data* - This is the protected/encrypted data [variable]

- *Padding* - To make sure that the data is the correct size for the cryptographic algorithm and ensure that the combination of the field's data, padding, padding length and next header are a multiple of four octets [variable]
- *Padding Length* - Number of octets of padding [1 octet]
- *Next Header/Protocol type* - same as protocol field in IPv4 or Next header in IPv6) [1 octet].
- *Authentication Data* - Cryptographic integrity check. The length is determined by the authentication function selected for the SA, if zero length, ESP is providing encryption only [variable]

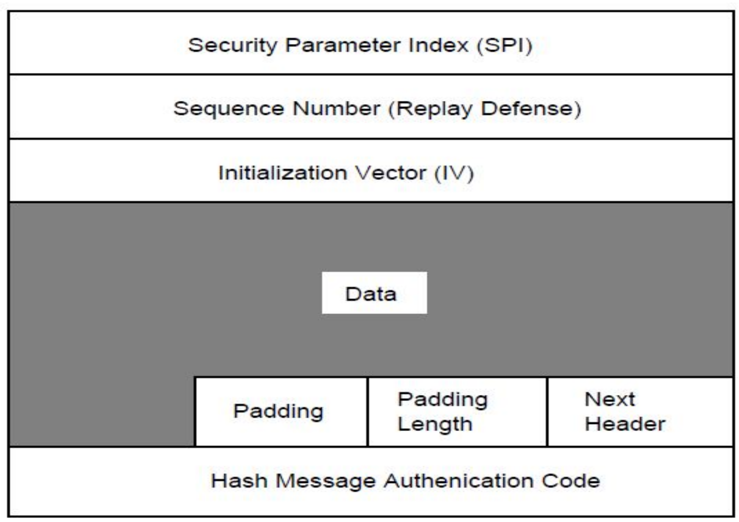

Figure 2.5: The ESP Header [14]

2.3.7 AH (Authentication Header)

AH [17] provides authentication only and if one wants encryption one must use ESP. Another difference between the ESP authentication and AH authentication is that AH also provides some protection of the IP header except for IP header field that can be modified by routers. For IPv4 AH the mutable fields are: Type of service, flags, fragment offset, TTL (Time to Live) and header checksum. The AH protocol protects the integrity of the IP datagram. To achieve this, the AH protocol calculates a HMAC to protect the integrity. When calculating the HMAC (Hashed Message Authentication Code) the AH protocol bases it on the secret key, the payload of the packet and the immutable parts of the IP header like the IP addresses. It then adds the AH header to the packet which is shown in Figure 2.6.

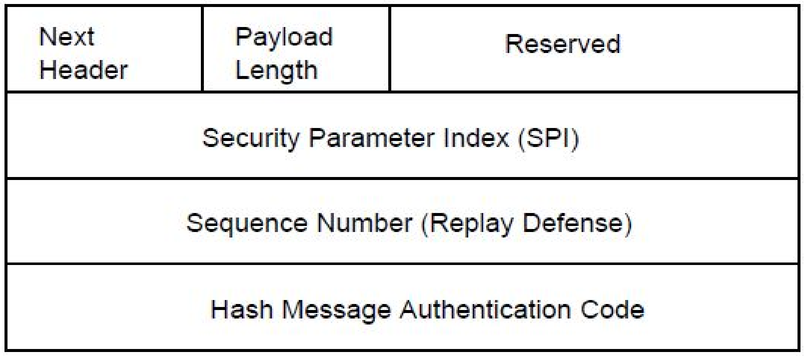

Figure 2.6: The AH Payload [14]

A brief description of 24 bytes long AH Header fields is as follows:

- *Next Header* - same as ESP [1 octet]
- *Payload Length* - The size of the AH header in 32bit chunks, not counting the first 8 octets [1octet]
- *Unused* [2 octets]
- *SPI* - same as ESP [4 octets]
- *Sequence Number* - same as ESP [4 octets]
- *Authentication data* - The cryptographic integrity check on the data/ HMAC calculated value [variable/maximum 12 octets]

2.4 MIKEY (Multimedia Internet KEYing)

MIKEY, explained in [18], is designed to meet the requirements of initiation of secure multimedia sessions. That is:

- The parameters for the security protocol should be exchanged in one round trip.
- The protocol should be simple and straight forward.
- The protocol should be possible to transport in session establishment protocols e.g. SIP-SDP.
- The protocol should supply end to end security for the keying material.
- Independence from any specific security functionality of the underlying transport.
- Low bandwidth consumption and low computational workload.

MIKEY is a keying protocol used to transfer, generate and negotiate cryptographic keys for multimedia applications. The most prominent mode of using MIKEY in a confined environment is Pre-shared key. The pre-shared key exchange is shown in Figure 2.7. The I_Message is the initiator's message which includes different payloads required for key negotiation. Similarly R_Message is the responder message contains verification messages as Verification Payload, Timestamp and Identification Payload.

MIKEY supports three types of key agreements:

1. Pre-shared key (PSK): In this method the pre-shared secret is used to derive keys both for encryption and integrity of the MIKEY message. In the MIKEY message the random generated TGK (Traffic encrypting key Generation Key) also called Master Key is securely transported. This is the most cost effective key agreement but the problem of distributing the shared secrets makes this solution hard to scale.

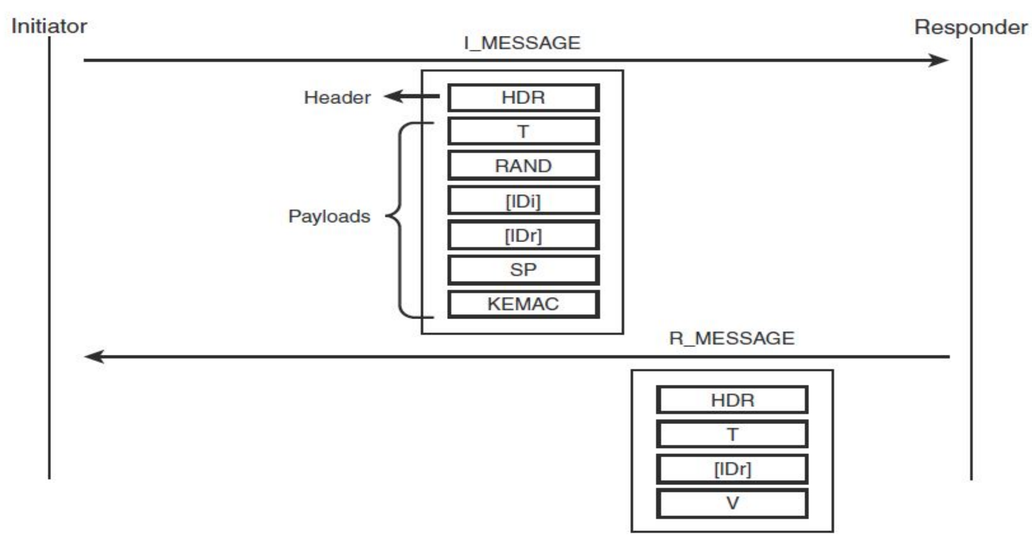

Figure 2.7: PSK message exchange in MIKEY (PSK message encapsulated in SDP attribute "a= keymgnt" [18]

Every payload has its own duty of work briefly stated below:

- *HDR* **-** The general MIKEY Header payload, which includes MIKEY CSB(crypto Session Bundle) related data (for example, CSB ID) and information mapping to the specific security protocol used like SRTP or IPSEC.

- *T* **-** The Timestamp payload, used mainly to prevent replay attacks.
- *IDx* **-** The Identity Payload of entity *x* (IDi = initiator, IDr = responder).
- *RAND* **-** Random/pseudo-random byte string also called nonce value, which is always included in the first message from the initiator. RAND is used as a freshness value for the key generation.
- *SP* **-** The Security Policies for the data security protocol (SRTP or IPSEC).
- *KEMAC* – It is a variable computed by encrypting all the TGK and using the predetermined MAC algorithm to provide integrity. The computation is as follows:

```
KEMAC = E(encr_key, {TGK}) || MAC
```

2. Public key encryption (PKE): Similar to PSK but the initiator makes a random key used for encryption and integrity. This key is then encrypted with the responder's public key and sent to the responder. This approach is more resource consuming but has the ability to scale if a Public Key Infrastructure (PKI) is available.

3. Diffie Hellman (DH): The only method that supplies perfect forward secrecy. This approach is more resource consuming and can only be used to establish single peer to peer keys. It also requires the existence of a PKI for message signing.

2.5 MIME (Multipurpose Internet Mail Extensions)

Multipurpose Internet Mail Extensions (MIME) defined in RFC2045 [19] and RFC2046 [20] was originally designed to extend the capabilities of mail bodies by introducing a body structure. This enabled the transfer of other content than plain US ASCII (American Standard Code for Information Interchange). MIME is now not only used to extend mail functionality but more as a general way of describing message bodies e.g. SIP-SDP.

MIME defines a number of new headers among others:

- MimeVersion:

- ContentType:
- ContentEncoding:
- ContentID:
- ContentDescription:

The most used SIP/MIME headers are:

- ContentLengh: describing the length of the message body in bytes
- ContentType: describing how the message body should be interpreted.

Example1:

ContentType: application/sdp

In the above row the content type is application and the applications subtype is sdp and the message body contains the Session Description Protocol (SDP). This is how SIP transports the SDP.

Example2:

ContentType: multipart/mixed; boundary=boun=_dry

The message body contains different body parts separated with the boundary boun=_dry.

2.6 S/MIME (Secure MIME)

S/MIME is defined in RFC 2633 [21]. S/MIME, as described in SIP's RFC, is used to secure the content of the SDP inside the SIP message. Note that use of S/MIME requires the existence of some PKI solution or preshared secrets. Example of two types of secure MIME bodies for SIP:

- multipart/signed: It is used for signing messages and signature is held separately from the SDP message so even a recipient that does not support S/MIME can read the message.
- application/pkcs7mime: It is used both for encrypted messages and signed messages. The message is first signed than encrypted. This way the identity of the signer is kept a secret. The message is encrypted as an enveloped message i.e. encrypted with a random session key that is protected by the public key of the recipient.

The following is an example of an encrypted S/MIME SDP body within a SIP message from [RFC 3261]:

```
----------------------------------------------------------------------------------------------------
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
CallID: a84b4c76e66710
CSeq: 314159 INVITE
MaxForwards: 70
Contact: <sip:alice@pc33.atlanta.com>
ContentType: application/pkcs7mime; smimetype=envelopeddata;
                  name=smime.p7m
ContentDisposition: attachment; filename=smime.p7m
                  handling=required
***************************************************
* ContentType: application/sdp                                *
*                                                             *
* v=0                                                         *
* o=alice 53655765 2353687637 IN IP4 pc33.atlanta.com         *
* s=                                                          *
* t=0 0                                                       *
* c=IN IP4 pc33.atlanta.com                                   *
* m=audio 3456 RTP/AVP 0 1 3 99                               *
* a=rtpmap:0 PCMU/8000                                        *
***************************************************
----------------------------------------------------------------------------------------------------
```

Where the SIP header fields:

- ContentType: indicates the media type of the message body sent to the recipient.
- ContentDisposition: describes how the message body is to be interpreted by the recipient.

CHAPTER 3

LITERATURE REVIEW

In the last decade, work on media session security has been done on the key exchange methods and ways to address drawbacks of already established MIKEY/SRTP standard. Exchanging keys and other cryptographic parameters is the most crucial and confidential part of any secure session. Traditional Key exchange mechanism available for IPSec is IKE. In multimedia, i.e. in VoIP, security is provided by SRTP which is an extended profile of RTP. Key exchange mechanism available for SRTP is MIKEY whose impact is described in [18, 22]. It confirms that the MIKEY can exchange keys and other cryptographic parameters in one round trip which is a huge advantage in multimedia application. In [23], the use of MIKEY keying protocol for parameters negotiation of SRTP and IPSec is presented. It is called as SIP initiated IPSec. The idea described in it is to transport all IPSec parameters in MIKEY message than Internet Keying exchange (IKE) method. It provides a good utilization of MIKEY protocol with the aggressive mode of Internet Key Exchange.

In the case of SRTP, the MIKEY message that contains SRTP's cryptographic parameters is carried within the SDP [24]. This is the most suitable option for SRTP as it is an application layer protocol and provides the confidentiality and authentication services at application layer. In the case of SIP initiated IPSec, the MIKEY message that contains IPSec's cryptographic parameters is not carried in SDP. SDP is not a good option in the case of SIP initiate IPSec because IPSec protects the traffic at the network layer and what traffic to protect is based on IP, transport protocol, ports and has nothing to do with the media session. IPSEC is independent from the media stream and one should be able to use both IPSEC protection as well as SRTP since they are independent of each other and add protection to the traffic at different layers. The SDP is used to describe media sessions. If one wants to protect just the traffic described in the SDP, that information can be inserted in the SDP. If the MIKEY message containing IPSec parameters is placed in the SDP, it than bounds to that specific media session and the flexibility of IPSEC is lost. Hence, the SDP is carried in SIP as a MIME message with content type *application/sdp* [20]. It then carries the SDP, with the media description as a multipart MIME body part, with content type field as *application/sdp* in a MIME multipart message with content type field as *multipart/mixed.* Within the multipart message, another body part contains a content type field as *application/mikey* containing the MIKEY message with necessary parameters for the IPSec setup. This is an efficient way in which the two body parts, *application/sdp* and *application/mikey*, meets the requirement of being independent. The research on security mechanism of SIP based VoIP systems [25]**,** had explained the confidentiality profile for RTP i.e. SRTP and authentication mechanism against spam [26].

So to retain the advantages offered by IPSec and SDP, the MIKEY message is carried in MIME payload in SIP i.e. SIP carry MIME payloads. The SDP with the media description is carried in a MIME message with ContentType application/sdp and the MIKEY message for IPSEC is carried in a ContentType application/mikey. The information in the body part application/mikey is a base64 encoded MIKEY message.. Using MIKEY for carrying IPSec parameters in SIP INVITE & 200OK response message provides a good utilization of Offer Answer model of SIP [27]. For utilizing SIP initiated IPSec, not only for VoIP session but for all IP traffic of the host, IPSec cryptographic parameters are transferred in separate MIME body. A separate MIME body for IPSec cryptographic parameters facilitates the use of S/MIME to secure call signaling. TLS is an option available for securing call signaling.

The Content-Type header gives power to the MIME encoded messages. Headers are used to specify the media type and subtype (separated by / symbol) of data in the body of a message. This header field identifies the media type of the message content. A media type consists of a type, a subtype, and one or more optional parameters. Contenttype:multipart headers identify multipart messages. The multipart/mixed content type is used when the body parts are independent and have to be bundled in a particular order. When a client does not recognize a multipart subtype, it will treat the message as multipart/mixed. Multipart/mixed specifies that the order of the body parts is important.

In the multipart content type, a boundary is present. The boundary tells the receiving client that the SIP message is divided into many parts and all those parts are separated by a string that is defined with *boundary*= keyword. A MIME-compliant client will only display or process content that follows the specified *boundary*= text strings. Boundaries are constructed by using the *boundary*= string, prepended by a double hyphen (--). The final body part is followed by the *boundary*= string with the double hyphen (--) appended on both sides of the keyword. Contenttype: application/mikey header tells that the content contains an Application type data i.e. data for a particular application with its subtype as MIKEY which describes the actual contents of the MIKEY message. Hence this part is containing the IPSEC parameters and Master key, Master Salt for IPSEC. Contenttype: application/sdp header, as described in [28], tells that it contains contents of Application type data i.e. for a particular application with its subtype as sdp which is the session description data sent in the INVITE for negotiation. In states that this part of MIME contains the SDP data used for negotiating media capabilities. The multipart/mixed header is used to describe that the different body parts should be handled independently. That is the key for using MIME or S/MIME so flexibly with MIKEY for secure parameter negotiation along with the Master Key.

Security at physical media communication infrastructure is one possibility for screening of spam and DOS attacks during the call establishment as presented in [29]. The call establishment as per paper "Call establishment delay for secure VoIP" [34], states that the establishment delays for a secure VoIP call (MIKEY/SRTP) is insignificant for a human user. The results were similar with MIKEY/IPSec-ESP in most aspects but with some differences in delays due to system calls. The system calls to set the IPSEC SA in the Linux IPSec implementation is very time consuming. But in case with MIKEY/IPSec-ESP, delays are longer due to system calls that set the IPSEC security associations and policy.

SIP initiated IPSec call establishment using MIKEY is shown in Figure 3. Eleven timestamps used in the client's softphone are shown as X1-X6 and Y1-Y5. ∂1 - ∂7 are the calculated time differences or delays measured as the time difference between the corresponding timestamps. The measurements recorded [34] lead to delays due to Kernel based cryptographic processing.

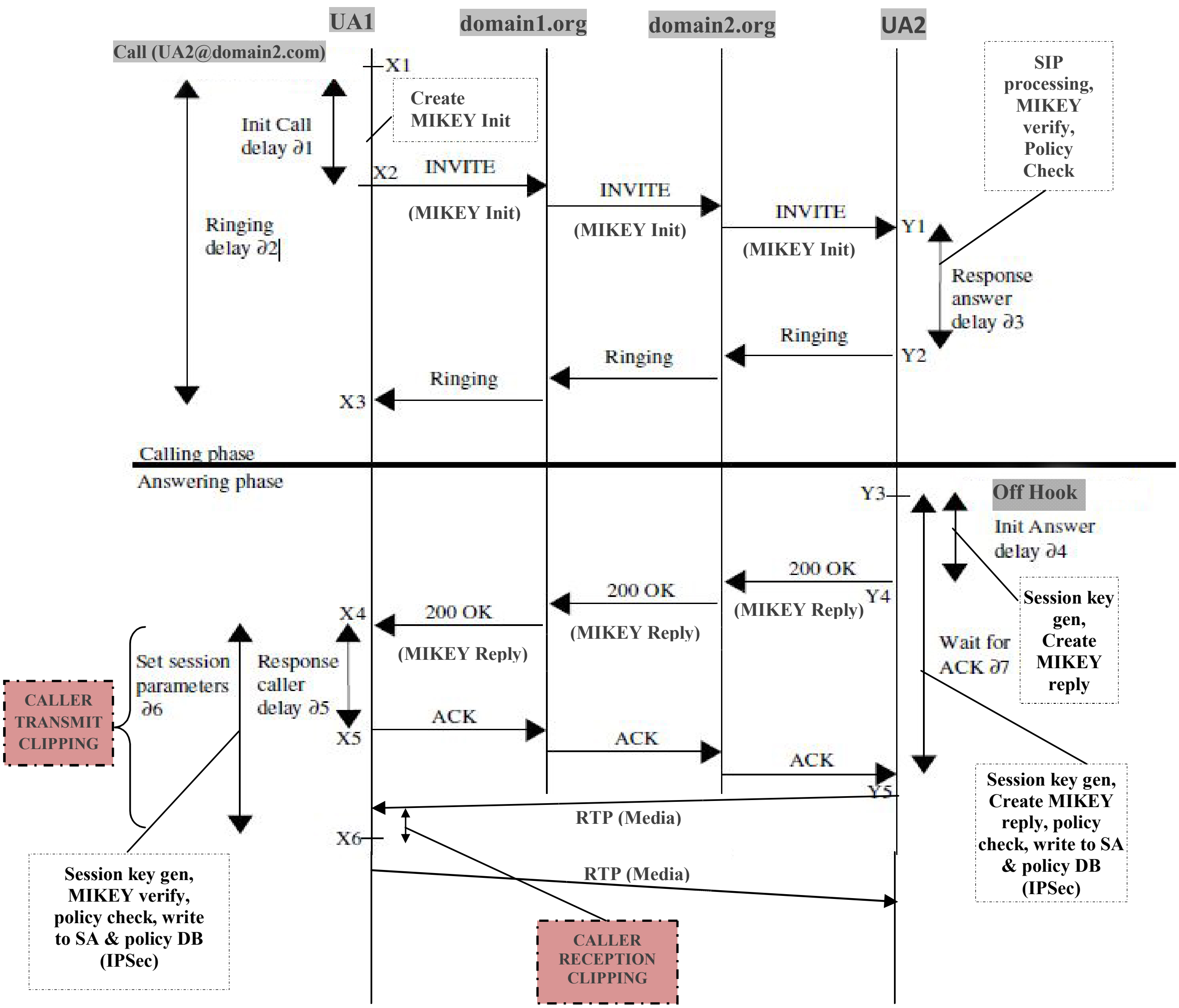

Figure 3: SIP's Secure Call establishment trapezoid with timestamps and calculated intervals [34]

$\partial 1$ is the time from the point when the caller makes the call and the creation of SIP and MIKEY Initiation message by UA1 softphone, to the point where the INVITE message (including MIKEY Initiation) leaves the UA1. $\partial 2$ (*Ringing delay)* is the time from when caller (UA1) has dialed callee's (UA2) number until she is notified that UA2 is ringing i.e. when caller receives the 180 Ringing Response from the callee. $\partial 3$ is the time it takes for the recipient / callee's user agent (UA1) to process the SIP INVITE, which also includes processing MIKEY for authentication and producing ringing at its end as well as producing 180 Ringing provisional messages for the caller. $\partial 4$ is the time from the point when the callee accepts the call to the point where the 200 OK messages leave the callee's user agent. This time includes the generation of Session keys. After generating Session keys, the callee's user agent create a MIKEY reply in which the chosen or successfully negotiated parameters (in success case) are indicated for IPSec such as ciphers, their modes, key length etc so to let it known by the Caller.

$\partial 5$ is the time it takes for the caller's user agent (UA1) to produce ACK after receiving and processing the 200 OK. It is important to note that the prerequisite for the generation of ACK message is to process the SIP 200 OK and the MIKEY authentication part only. There is no need to wait for generation of session keys and IPSec SA and Policy writing to database. $\partial 6$ is the time it takes for the caller's user agent (UA1) to process the 200 OK, MIKEY authentication, generation of ACK message, setting the session parameters and writing to IPSec SA & Policy Database. More precisely, it is the time when UA1 has received the 200 OK SIP response, but is unable to transmit due to ongoing cryptographic processing i.e Session key generation, Writing to SA & Policy Database. This is *caller transmit clipping* and *caller reception clipping.* $\partial 7$ *(Answering delay)* is the time the callee's user agent (UA1) waits until receiving the ACK. This is the answering delay. It is the Time when UA2 has picked up the phone, but is unable to transmit due to ongoing cryptographic processing. We refer to this as the *callee transmit clipping*. It also includes the same processing time with respect to number of activities it has to perform such as Session key generation, writing to SA and Policy database for IPSec.

$\partial 4$ and $\partial 6$ ($\partial 7$) in MIKEY/IPSec-ESP includes the setting of SA to the kernel and 2 inclusive function calls to the function that adds the SA and policy to IPSec Database. Each function call takes about 330mS [34]. This value would be different on a faster computer but would still measure in tenth of milliseconds.

The caller sends the MIKEY Initiation in its SIP INVITE message to Callee. The Callee on the other hand sends back the negotiated parameters in the 200OK final response message which is created and sent by Callee only when he picks up the phone. This MIKEY response contains negotiated IPSec parameter. The caller after processing SIP part of the 200OK response sends back the acknowledgement ACK immediately and starts its cryptographic processing of session key generation and setting the Security associations and policy. This implies that all of the cryptographic processing is done on callee end when he picks up his phone. The cryptographic processing includes:

- Creation of MIKEY reply for conveying the supported or negotiated cryptographic parameters.
- Generation of Cryptographic keys such as Session keys for both the Outbound and Inbound traffic.
- Creation of SAs (Security Associations) as per the proposal and negotiated parameters in the SAD (Security Association Database).
- Creation of SPs (Security Policies) in the SPD (Security Policy Database).

This high amount of cryptographic processing at the caller end leads to problem of *clipping effects* and *ghost ringing* which are described in [34]. These two problems are the byproducts of SIP initiated IPSec method. The high amount of processing by caller in the answering phase when the receiver picks up the phone, makes the user unable to transmit their voice packets i.e. RTP packets. As the caller sends back the ACK message immediately after receiving the 200OK MIKEY reply, the propagation delay of ACK message and the ACK processing is lesser than the IPSec cryptographic processing time of the caller. Due to this, the receiver attains ready to transmit state earlier than the caller which is still bust in cryptographic processing. When receiver send the RTP packets, it gets droped by the caller due to ongoing cryptographic processing. This is called the *call reception clipping*, an inability to process voice packets. The clipping effects that the caller faces are *the call reception clipping* and *the call transmit clipping*. It means that, for the first few hundreds of microseconds, the caller discards all the RTP media packets coming from the responder as well as not able to transmit any of the voice packets. This happens because of the ongoing cryptographic processing at caller's operating system. Although SIP initiated IPSec has an advantage of establishing IPSec tunnel between two hosts but not only for Voice packets, but to general IP traffic also.

In the present research, a new algorithm has been proposed to eliminate the clipping effects and ghost ringing problems. This solution is based on the reliable provisional acknowledgement of unreliable 1xx responses using PRACK as described in [10]. To accomplish this, it requires both the User Agents (UA) to support PRACK and the solution also describes the successful and unsuccessful scenarios of PRACK support.

CHAPTER 4

PROBLEM STATEMENT

The call flow presented in previous and its results states that the existing SIP-MIME-MIKEY infrastructure for IPSEC is posing delays at the *answering phase* which are further leading to problems of *call clippings* at the start of the *Media Session*. The values of ∂7 - ∂4 (∂7 minus ∂4) needs to be greater than ∂6 otherwise the ACK would reach the callee (UA2) before the caller (UA1) has set the session parameters. As the caller sends back the ACK immediately when caller receive the 200 OK message containing MIKEY Reply and by processing SIP headers and MIKEY authentication. This results in the loss of RTP packets in the beginning of the call known as call clipping.

4.1 MIKEY Reply in 200 OK

The most straightforward way described in the Literature Review is to put the *MIKEY Reply* in the *200 OK final response* when Callee (UA2) picks up his phone. This approach is shown in Figure 4.1. The SIP message exchange is the same as in the regular SIP call establishment.

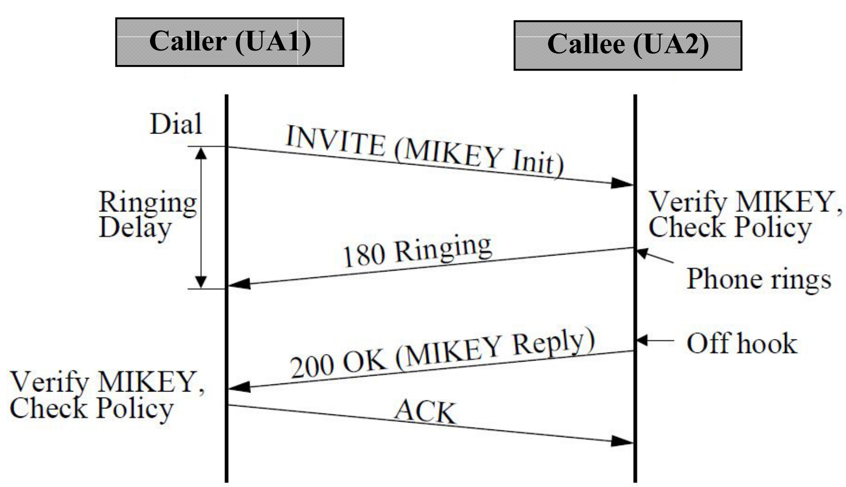

Figure 4: MIKEY request in SIP INVITE and reply in 200 OK (An existing Design) [26]

The caller sends the MIKEY init message in SIP INVITE which is verified at the Recipient end where on reception and processing of SIP INVITE and MIKEY authentication, a provisional (no ACK for this response) response as *180 Ringing* is sent back. This provisional Ringing message enables the caller to know reception of the INVITE message and the start of the Ringing.

The reason the *200 OK* (and *not* the *180 Ringing* which is one solution to this thesis) is used to carry the *MIKEY Reply* is that the *200 OK response* is sent reliably as opposed to the 180 Ringing provisional response. This mechanism is used when if either the Caller (UA1) or Callee (UA2) does not support PRACKs (provisional acknowledgements) [10].

4.2 Call establishment Delay in MIKEY / IPSec

In the case of MIKEY / IPSec-ESP, values $\partial 7$ - $\partial 4$ ($\partial 7$ minus $\partial 4$) should be greater than $\partial 6$. Hence the ACK reaches the callee (UA1) before the caller (UA1) has set the session parameters every time due to system calls to the operating system kernel space The IPSEC is closely dependent on and bounded to the operating system because writing to IPSec SA & policy database requires update from the system calls to the kernel. These function calls requires time which results in the loss packets in the beginning of the call for the duration of 0.7s (700ms). It means, when the callee answers the call, there will be no one on the other end for the first 0.7 seconds.

Although, simplicity is an advantage of this solution but it also suffers from excessive delays and its affect. Note that UA2 is able to filter VoIP spam calls, since caller authenticates the callee and checks the call against the policy configuration *before* the phone rings. The problems are:

1. Some MIKEY and SRTP/IPSec setup processing takes place after Callee (UA2) picks up the phone, which can lead to *clipping* effects at the beginning of the call.

2. Callee (UA2) may suffer from *ghost ringing* if Caller (UA1) for some reason rejects the *MIKEY Reply*, and thus the call. Although the probability of occurring the latter case is less, it still is viable to happen in a situation when Caller's policy check does not accept the credentials which Callee presents. (If Caller wants to reject a call before it has been established, it should be done by sending a CANCEL message to Callee.)

The problems that are produced as raw effects of delays are discussed below:

1. **Clipping Effects:**
   - *Callee Transmit Clipping* **(b2)**: Time when Callee (UA2) has picked up the phone, but is unable to transmit due to ongoing cryptographic processing.
   - *Caller Transmit Clipping* **(a2)**: Time when Caller's user agent (UA1) has received the 200 OK SIP response, but is unable to transmit due to ongoing cryptographic processing as Caller UA1 may try to talk as soon as the local ring tone is stopped but caller is not able to do so due to ongoing cryptographic processing.
   - *Caller Reception Clipping* (**a4):** Time when Caller receives data from Callee, but is unable to process it due to ongoing cryptographic processing.

The measured clipping delays at both the caller and the callee are much higher when using IPSec than when using SRTP as explained in Literature. System calls setting the security association and the IPSec policy are responsible for the main part of this extra delay.

2. **Ghost Ringing:**

   If caller for some reasons reject the MIKEY reply, the Callee have no mechanism in current implementation for getting notified about the rejection of his MIKEY Reply before *180 Ringing* provisional message. The Callee in this case suffer from Ghost Ringing which means that nobody will be on the Caller side when Callee picks up the phone because Callee's phone start ringing immediately when after sending the *180 Ringing* provisional message to Caller. So there should be a mechanism that enables the Caller and Callee to get their Local Ringing tone after all the policy checks at both the ends.

So the current SIP-MIME-MIKEY infrastructure for IPSec establishment suffers from these two ill effects in which first one (Clipping Effects) can decline the QOS at the beginning of the call and the latter one provides a problem called Ghost Ringing i.e. ringing of receiver's phone while caller has rejected the mikey reply.

Therefore, the objectives of this thesis can be outlined as:

- To Study and analyze the Existing design.
- To create a Solution Design that removes the Call Clipping and Ghost Ringing.

CHAPTER 5

SOLUTION DESIGN

To avoid *clipping effects* due to cryptographic processing at the start of the media session and to eliminate the risk of *ghost ringing*, this thesis provides a new solution design in which SIP phones implements support for reliable provisional acknowledgments (PRACKs). Due to cryptographic processing delays, because of MIKEY Reply in 200OK response message, the clipping of media packets (RTP packets) occurs at the beginning of the media session. The solution is to send the MIKEY reply earlier than the 200 OK response (before Receiver/Callee picks up the phone) i.e. to send the MIKEY reply in the *calling phase* rather than in *answering phase*.

To achieve this objective, the primary condition required to fulfill is:

$$(\partial 7 - \partial 4) > \partial 6 \qquad \text{.....} (1)$$

Where: **∂6** is the processing delay by the caller after receiving the 200OK. It is the time it takes to process the 200OK reply (may contain MIKEY reply), send back the *ACK* message and setting up the security associations and policy.

∂7 is the *answering delay* of callee i.e. processing time it takes when callee picks up the phone, sends 200OK and setup security associations and policy to the time it takes to receive the *ACK* message from the caller.

∂4 is the *Initial answer delay* by the caller i.e. processing time it takes when callee picks up the phone to the time it sends 200OK response.

Equation (1) holds well when *'no security'* processing is involved in answering phase as described in the Literature Review. Equation (1) is satisfied by removing the cryptographic processing in the answering phase. This enables the SIP phone to perform only the general SIP processing in the *answering phase* which doesn't include any cryptographic processing delays.

So, MIKEY reply can be sent in:

1. MIKEY Reply in 100 trying (which is not a valid option, because of absence of standard support)
2. MIKEY Reply in 180 Ringing Response.
3. MIKEY Reply in 183 Session in Progress Response.

5.1 Solving Problem of Delays (Clipping Effects & Ghost Ringing)

In the Table 2, the various configuration combinations for the User Agent Client (UAC) and User Agent Server (UAS) are presented. The PRACK support and requirement enables the PRACK call else the normal call is automatically chosen. Hence in this solution, it is the PRACK call that is a prerequisite for this solution.

Table 5: The call behavior of UAC & UAS with various configuration combinations

UAC	UAS	Call Processing
PRACK Disabled	PRACK Disabled	Normal Call
PRACK Disabled	PRACK Supported	Normal Call
PRACK Disabled	PRACK Require	Call Rejected
PRACK Supported	PRACK Disabled	Normal Call
PRACK Supported	PRACK Supported	PRACK Call
PRACK Supported	PRACK Require	PRACK Call
PRACK Require	PRACK Disabled	Normal Call
PRACK Require	PRACK Supported	PRACK Call
PRACK Require	PRACK Require	PRACK Call

5.1.1 MIKEY Reply in 180 Ringing

If Caller (UA1) announces PRACK (Provisional Acknowledgement) support in his SIP INVITE message, than Callee (UA2) can send the *MIKEY Reply* in the 180 Ringing message as mentioned in table 5.1 and presented in figure 5.1. Caller can send the

PRACK (Provisional Acknowledgement) immediately after authenticating the MIKEY message. The various processing delays are labeled as ∂ and explained are below:

$\partial 1$: The time from the point when the caller makes the call and the creation of SIP and MIKEY Initiation message by UA1 Softphone, to the point where the INVITE message leaves the UA1.

$$\Rightarrow \qquad \partial 1 = X2 - X1$$

Where X1: Timestamp when UA1 dials the phone

X2: Timestamp when UA1 completes the creation of SIP INVITE.

$\partial 2$: The time UA1 takes in processing the SIP 180 Ringing reliable provisional response. It includes the SIP verification, MIKEY verification, Policy check processing time and the creation time of PRACK message as per Algorithm in section 5.2.1.1.

$$\Rightarrow \qquad \partial 2 = X4 - X3$$

Where X3: Timestamp when UA1 receives the 180 Ringing Reliable provisional response.

X4: Timestamp when UA1 completes the creation of SIP verification, MIKEY verification, policy check processing time and the creation time of PRACK message.

$\partial 3$: This is the Ringing Delay which now includes the MIKEY Processing delay. This is the time from when Caller (UA1) dials Callee's (UA2) number until she is notified that UA2 is ringing. i.e when Caler receives & process the 180 Ringing Reliable Provisional Response from the Callee.

$$\Rightarrow \qquad \partial 3 = \partial 1 + \partial 2 + \partial 4 + \text{Network dependent propagation delay}$$

$\partial 4$: The time UA2 (Recipient) takes in processing the SIP INVITE. It includes the SIP verification, MIKEY verification, Policy check processing time, Creation of MIKEY Reply and the creation time of 180Ringing message as per Algorithm in section 5.2.1.2. This is the most time consuming process at the recipient end.

$$\Rightarrow \qquad \partial 4 = Y2 - Y1$$

Where Y1: Timestamp when UA2 receives the SIP INVITE message.

Y2: Timestamp when UA2 completes the SIP verification, MIKEY verification, Policy check processing time, Creation of MIKEY Reply and the creation of 180Ringing message.

$\partial 5$: The time UA2 takes in processing the PRACK till the creation of final 200OK response for the PRACK. It doesn't include the Session key generation, writing to SA & Policy database of IPSec. Although all the latter mentioned processes starts immediately after the creation process of 200OK for PRACK.

$$\Rightarrow \quad \partial 5 = Y4 - Y3$$

Where Y3: Timestamp when UA2 receives the PRACK message.

Y4: Timestamp when UA2 completes processing of PRACK message and the creation of 200OK message for PRACK.

$\partial 7$: It is the Time when UA2 has picked up the phone till UA2 receive the ACK. This is the answering delay. , It includes the time the callee's user agent (UA2) take in creation of 200OK response for INVITE till the reception of Acknowledgement.

$$\Rightarrow \quad \partial 7 = Y7 - Y5$$

Where Y5: Timestamp when UA2 creates 200OK message for SIP INVITE.

Y7: Timestamp when UA2 completes processing of ACK message.

$\partial 6$: This is the response caller delay. It includes the processing of 200OK INVITE & creation of Acknowledgement message for the UA2.

$$\Rightarrow \quad \partial 6 = X6 - X5$$

Where X5: Timestamp when UA1 process the 200OK message of SIP INVITE.

X6: Timestamp when UA1 create the ACK message for UA2.

$\partial 8$: This is the initial answer delay. It is the time when the callee accepts the call to the point where 200OK final response leaves the message.

$$\Rightarrow \quad \partial 8 = Y6 - Y5$$

Where Y5: Timestamp when UA2 creates 200OK message for SIP INVITE.

Y6: Timestamp when UA2 sends the ACK message for UA1.

Inferences:

In this solution design, Answering Delaying is reduced to the case of *no security*, in which the condition of equation (1) holds perfectly well i.e. ($\partial 7$-$\partial 8$) > $\partial 6$ in this case as shown in Figure 5.1. This solution design has removed the cryptographic processing from the answering delay. The answering phase now has the delay condition of equation (1) satisfied. This logically proves the elimination of cryptographic delays from the *answering phase* and increase in ringing delay in *ringing phase* which has no effect on the quality of call.

As compared to the prior case (MIKEY reply in 200OK response message), Caller (UA1) experience a somewhat longer *Ringing delay*, since Callee (UA2) has to construct the *MIKEY Reply* before sending the 180 Ringing message. This does not constitute any problem since Caller is less sensitive to delays before caller gets a local ringing tone as opposed to clipping effects at the beginning of the call.

If Caller wants/needs to rejects the *MIKEY Reply,* than caller has to send a CANCEL message to Callee (instead of sending the PRACK). As in this solution, Callee's phone start ringing immediately after sending the *MIKEY Reply,* this solution still suffers from *ghost ringing* whose solution design is presented in section 5.2.2.

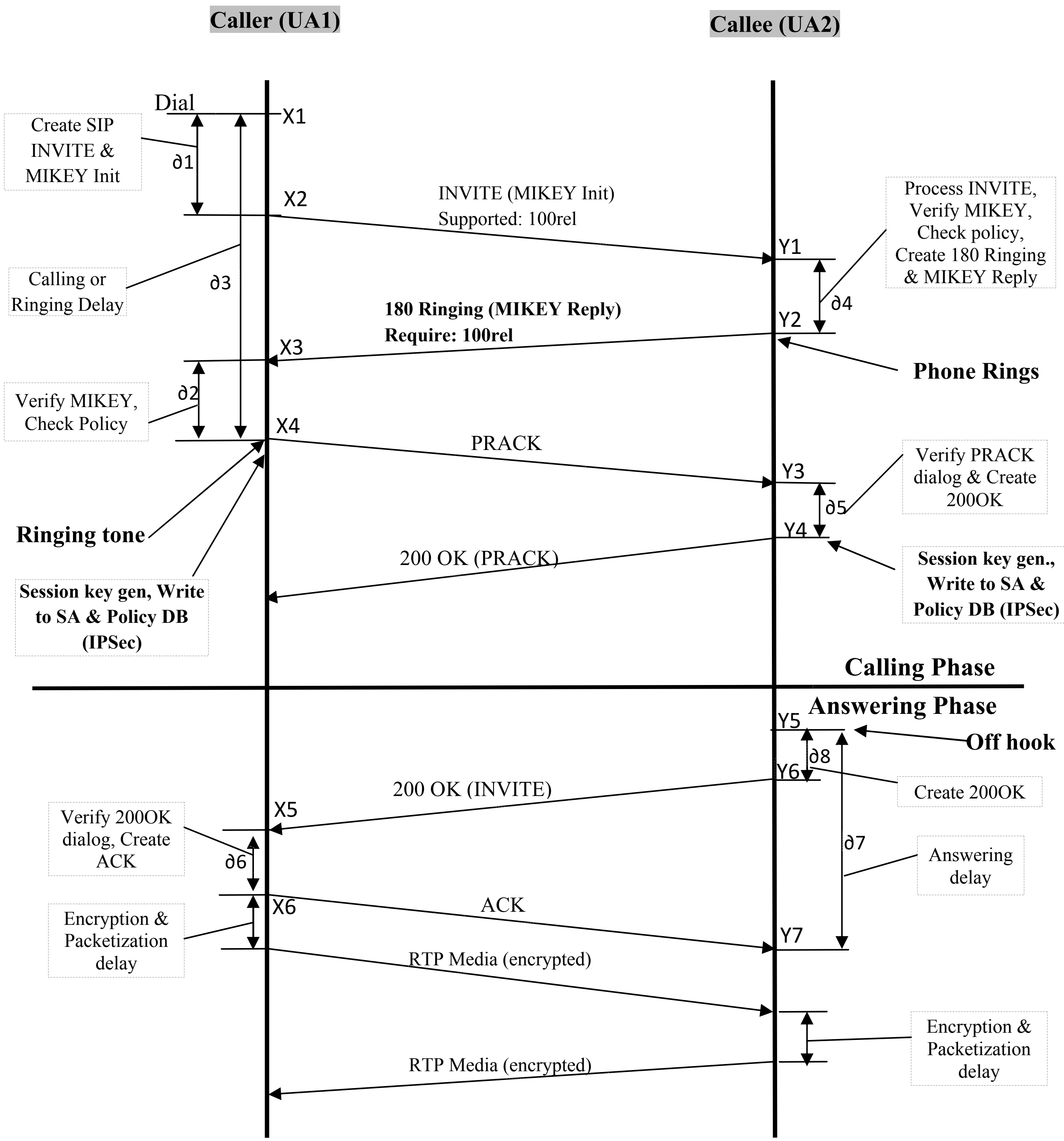

Figure 5.1: MIKEY Request in SIP INVITE & Reply in Provisional 180 Ringing

Format of SIP INVITE with respect to figure 5.1 is given below:

INVITE sip: UA2@domain2.org; user=phone SIP/2.0
From: <sip: UA1@domain1.org; user=phone>;tag=(*tag_number)*
To: <sip: UA2@domain2.org; user=phone>
Via: SIP/2.0/UDP UA1pc.domain2.org:port_number;
CallID: 196120334@ UA1pc.domain2.org
CSeq: *(Call sequence number)* INVITE
Supported: 100rel
Contact:<sip:UA1@UA1pc.domain1.org:port_number;user=phone;
transport=UDP>;
expires=(*expiry time)*
UserAgent: Minisip
ContentType: multipart/mixed; *boundary=boun=_dry*
ContentLength: (*Total length of contents below)*

SIP

boun=_dry
Contenttype: application/mikey

/ contains IPSec Cryptographic parameters, MIKEY authentication key and its parameters in byte64 format */*

MIKEY's
MIME
Body

--boun=_dry
Contenttype: application/sdp

v=
o=
s=
c=
t=
m=
a=

SDP's
MIME
Body

--boun=_dry—

Format of SIP 180 Ringing with respect to figure 5.1 is given below:

SIP/2.0 180 Ringing
From: <sip: UA1@domain1.org; user=phone>;tag=(*tag_number)*
To: <sip: UA2@domain2.org; user=phone>
Via: SIP/2.0/UDP UA1pc.domain2.org:port_number;
CallID: 196120334@ UA1pc.domain2.org
CSeq: *(Call sequence number)* INVITE
Require: 100rel
Contact:<sip:UA1@UA1pc.domain1.org:port_number;user=phone;transport=UDP>;
expires=(*expiry time)*
UserAgent: Minisip
ContentType: multipart/mixed; *boundary=boun=_dry*
ContentLength: (*Total length of contents below)*

SIP

boun=_dry
Contenttype: application/mikey

/ contains negotiated IPSec Cryptographic parameters, MIKEY authentication key and its parameters in byte64 format */*

MIKEY's MIME Body

--boun=_dry
Contenttype: application/sdp

v=
o=
s=
c=
t=
m=
a=

--boun=_dry—

SDP's MIME Body

5.1.1.1 Algorithm for UAC

Step 1- Create SIP INVITE message and insert Supported: 100rel in SIP header using the MIME structure with a separate body for MIKEY containing IPSec parameters in MIKEY Init Message.

Step 2- Prepare to send the SIP INVITE using DNS query or resolved host address and send it.

Step 3- Check whether the received provisional response message is from valid User and in the same dialog of SIP INVITE.

Step 4- If the response message is from valid user and same dialog, than

Step 4.1 Process the MIKEY Payloads for authentication and policy check and copy the RSeq number.

Step 4.2 Else discard the response and wait for valid response till the timer expires

Step 5- If the 180 Ringing response doesn't contain Support/Require Header in SIP, than

Step 5.1- Exit this algorithm and call algorithm (Mikey reply in 200 OK)

Step 5.2- Else, create and send the PRACK message with RAck = RSeq, set with acknowledgement number equals to RSeq received in the 180 Ringing provisional response.

Step 6- Start the Local Ringing tone.

Step 7- Generate Session key from the master key using MIKEY.

Step 8- Write the Security Association and policy received in the SA & Policy databases of IPSec by the system calls to kernel space.

Step 9- Resend the PRACK until 200 OK (PRACK) is received till timeout.

Step 10- If timeout expires, than

Step 10.1- Disconnect the call and generate PRACK unacknowledged error.

Step 10.2- Else Process the SIP final response to INVITE, 200 OK (INVITE) and stop the Local Ringing tone

Step 11- Create and send the ACK, acknowledging the Callee about the reception of the 200 OK final response.

Step 12- Create and send the RTP media packets as per security association and policy established.

5.1.1.2 Algorithm for UAS

Step 1- Process the SIP INVITE message received for the identity verification/spam.

Step 2- Process the MIKEY payload for authentication and policy check.

Step 3- If MIKEY authentication fails, than

Step 3.1- Drop the SIP INVITE message and create & send the 6xx error message to caller.

Step 3.2- Else Process the MIKEY & inspect the IPSec ciphers support at UAS Application and create the MIKEY reply with respect to supported features and policy adopted (Parameter negotiation).

Step 4- If Require header field is present with tag Require= 100req, than

Step 4.1- Create and send 180 Ringing response with Require= 100rel and a new field RSeq for sequencing to an initial value indicating Reliable provisional requirement and insert MIKEY Reply in MIME body part.

Step 4.2- Else Create and send 180 Ringing provisional response without MIKEY reply and call algorithm (Mikey reply in 200OK) and exit from this algorithm.

Step 5- Start the Phone Ringing.

Step 6- Process the PRACK and check for the RAck value for being in the same dialog else drop the call.

Step 7- Create and send 200 OK for PRACK.

Step 8- Generate Session Key from the master key received using MIKEY.

Step 9- Write the Security Association and policy received in the SA & Policy databases of IPSec by the system calls to kernel space.

Step 10- If the User Agent accept/picks the call (OFF the hook), than

Step 10.1- Create and send the SIP response 200OK (INVITE).

Step 10.2- Else create and send BYE after timeout.

Step 11- Process the ACK message, create and send the RTP & RTCP media packets.

5.1.2 MIKEY Reply in 183 Session in Progress:

The Ghost Ringing is a situation in which nobody is on the Caller side when Callee picks up the Ringing phone. This happens because Callee's phone starts ringing immediately after sending the 180 Ringing provisional message to Caller. If, for any reason, caller rejects the Mikey reply in 180 Ringing, the Callee won't be able to ignore the Ringing that started immediately after sending the 180 Ringing message.

So there should be a mechanism that enables the Caller and Callee to get their Local Ringing tone after all the policy checks at both the ends. To eliminate the problem for *Ghost ringing,* this design sends *MIKEY Reply* in 183 Session in Progress provisional reliable response message. The Callee's phone rings first when the corresponding PRACK arrive as depicted in the Figure 5.2. Although this increases the *Ringing delay* even more, but this is the most appropriate way to send the *MIKEY Reply* that eliminates the two ill-effects of delays i.e. Clipping and Ghost Ringing. In order for this approach to work, the Caller has to wait until it has accepted Callee's reply before it sends back the PRACK (Caller rejects the message by sending a CANCEL message). The process is all same except the introduction of new 183 Session in Progress provisional reliable response message and delay of the function call to local ringing tone at both the ends. To solve this problem of Ghost Ringing as well as the Clipping Effect, there are two conditions to satisfy:

1. Equation (1) (same as that of Solution design, mikey reply in 180 ringing)
2. Call to ringing function after the creation and reception of 180 ringing provisional message.

The various processing delays are labeled as ∂ and explained below:

$\partial 1$: The time from the point when the caller makes the call and the creation of SIP and MIKEY Initiation message by UA1 Softphone, to the point where the INVITE message leaves the UA1.

$$\Rightarrow \quad \partial 1 = X2 - X1$$

Where X1: Timestamp when UA1 dials the phone

X2: Timestamp when UA1 completes the creation of SIP INVITE.

$\partial 2$: The time UA1 takes in processing the SIP 183 Session in Progress reliable provisional response. It includes the SIP verification, MIKEY verification, Policy check processing time and the creation time of PRACK message as per Algorithm in section 5.2.1.1.

⇨ $\partial 2 = X4 - X3$

Where X3: Timestamp when UA1 receives the 183 Session in Progress Reliable provisional response.

X4: Timestamp when UA1 completes the creation of SIP verification, MIKEY verification, policy check processing time and the creation time of PRACK message.

$\partial 3$: This is the Ringing Delay which now includes the PRACK processing time & Creation of 200OK for PRACK. This is the time from when Caller (UA1) dials Callee's (UA2) number until she is notified that UA2 is ringing i.e. when Caller receives & process the 183 Session in Progress Reliable Provisional Response from the Callee.

⇨ $\partial 3 = \partial 1 + \partial 2 + \partial 4 + \partial 5 + \partial 8$ + N/W dependent propagation delay

$\partial 4$: The time UA2 (Recipient) takes in processing the SIP INVITE. It includes the SIP verification, MIKEY verification, Policy check processing time, Creation of MIKEY Reply and the creation time of 183 Session in Progress message as per Algorithm in section 5.2.2.2.

⇨ $\partial 4 = Y2 - Y1$

Where Y1: Timestamp when UA2 receives the SIP INVITE message.

Y2: Timestamp when UA2 completes the SIP verification, MIKEY verification, Policy check processing time, Creation of MIKEY Reply and the creation of 183 Session in Progress message.

$\partial 5$: The time UA2 takes in processing the PRACK till the creation of final 200OK response for the PRACK. It doesn't include the Session key generation, writing to SA & Policy database of IPSec. Although all the latter mentioned processes starts immediately after the creation process of 200OK for PRACK.

⇨ $\partial 5 = Y4 - Y3$

Where Y3: Timestamp when UA2 receives the PRACK message.

Y4: Timestamp when UA2 completes processing of PRACK message and the creation of 200OK message for PRACK.

$\partial 6$: It is the Time when UA2 has picked up the phone till UA2 receive the ACK. This is the answering delay. , It includes the time the callee's user agent (UA2) take in creation of 200OK response for INVITE till the reception of Acknowledgement.

⇨ $$\partial 6 = Y7 – Y5$$

Where Y5: Timestamp when UA2 creates 200OK message for SIP INVITE.

Y7: Timestamp when UA2 starts processing of ACK message.

$\partial 7$: This is the response caller delay. It includes the processing of 200OK INVITE & creation of Acknowledgement message for the UA2.

⇨ $$\partial 7 = X6 – X5$$

Where X5: Timestamp when UA1 process the 200OK message of SIP INVITE.

X6: Timestamp when UA1 create the ACK message for UA2.

$\partial 8$: This is the time used to create provisional unreliable 180 Ringing message. It also includes the processing time of function of local Phone Ringing.

⇨ $$\partial 8 = Z – Y4$$

Where Z: Timestamp when UA2 create the 180 Ringing message & process the local Phone Ringing function.

Y4: Timestamp when UA2 completes processing of PRACK message and the creation of 200OK message for PRACK.

$\partial 9$: This is the initial answer delay. It is the time when the callee accepts the call to the point where 200OK final response leaves the message.

⇨ $$\partial 8 = Y6 – Y5$$

Where Y5: Timestamp when UA2 creates 200OK message for SIP INVITE.

Y6: Timestamp when UA2 sends the ACK message for UA1.

Inferences:

The two necessary conditions for elimination of *Ghost Ringing and Clipping effects* are satisfied. This logically proves the elimination of cryptographic delays from the *answering phase* and increase in ringing delay in *ringing phase* which has no effect on the quality of call. This solution design is an add on version of solution presented in 5.2.1, which not only removes the Clipping effects due to IPSec Cryptographic processing but also remove the Ghost Ringing. In this solution, Caller (UA1) experiences a somewhat longer r*inging delay*. This is because this solution has delayed the 180 Ringing message after all the policy verification. This does not constitute any problem too since Caller is less sensitive to delays before it gets a local ringing tone as opposed to clipping effects at the beginning of the call. In this solution design, Answering Delaying remains same as solution design presented in MIKEY Reply in 180 Ringing. The Equation (1) still holds perfectly.

Format of SIP INVITE with respect to figure 5.2 is given below:

```
-----------------------------------------------------------------------------------------------------------
INVITE sip: UA2@domain2.org; user=phone SIP/2.0
From: <sip: UA1@domain1.org; user=phone>;tag=(tag_number)
To: <sip: UA2@domain2.org; user=phone>
Via: SIP/2.0/UDP UA1pc.domain2.org:port_number;
CallID: 196120334@ UA1pc.domain2.org
CSeq: (Call sequence number) INVITE
Supported: 100rel
Contact:<sip:UA1@UA1pc.domain1.org:port_number;user=phone;
transport=UDP>;
expires=(expiry time)
UserAgent: Minisip
ContentType: multipart/mixed; boundary=boun=_dry
```

SIP

```
ContentLength: (Total length of contents below)

boun=_dry
Contenttype: application/mikey

/* contains IPSec Cryptographic parameters, MIKEY authentication
 key and its parameters in byte64 format */
```

MIKEY's MIME Body

```
--boun=_dry
Contenttype: application/sdp
v=
o=
s=
c=
m=
a=
--boun=_dry—
-----------------------------------------------------------------------------------------------------------
```

SDP's MIME Body

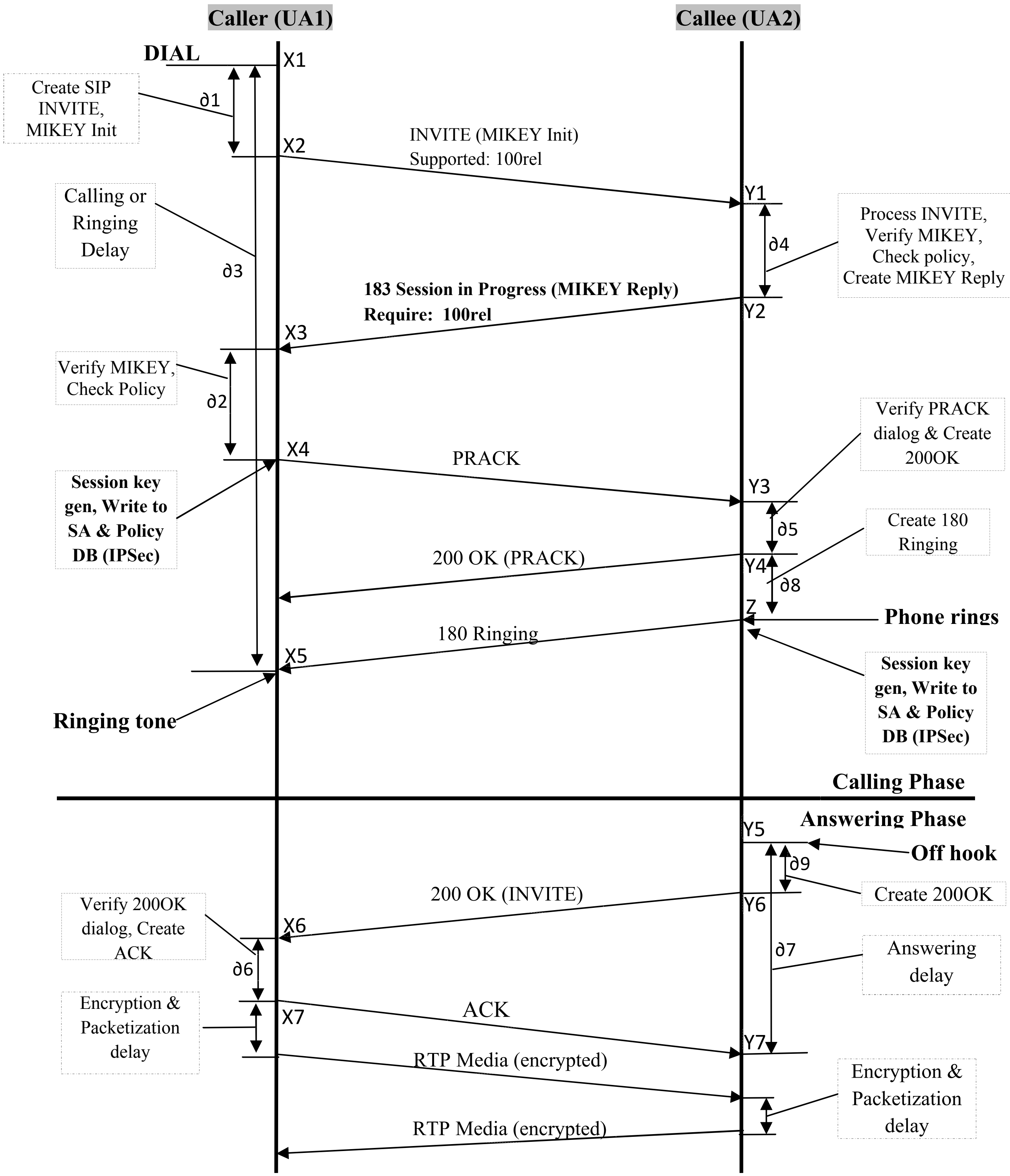

Figure 5.2: MIKEY Request in SIP INVITE & Reply in Provisional 183 Session in Progress

Format of SIP 183 Session in Progress with respect to figure 5.1 is given below:

SIP/2.0 183 Session In Progress
From: <sip: UA1@domain1.org; user=phone>;tag=(*tag_number)*
To: <sip: UA2@domain2.org; user=phone>
Via: SIP/2.0/UDP UA1pc.domain2.org:port_number;
CallID: 196120334@ UA1pc.domain2.org
CSeq: *(Call sequence number)* INVITE
Require: 100rel
Contact:<sip:UA1@UA1pc.domain1.org:port_number;user=phone;
transport=UDP>;
expires=(*expiry time)*
UserAgent: Minisip
ContentType: multipart/mixed; *boundary=boun=_dry*
ContentLength: (*Total length of contents below)*

SIP

boun=_dry
Contenttype: application/mikey

/ contains negotiated IPSec Cryptographic parameters,*
*MIKEY authentication key and its parameters in byte64 format */*

MIKEY's MIME Body

--boun=_dry
Contenttype: application/sdp

v=
o=
s=
c=
m=
a=

SDP's MIME Body

--boun=_dry—

5.1.2.1 Algorithm for UAC

Step 1- Create SIP INVITE message and insert Supported: 100rel in SIP header in the MIME structure with a separate body for MIKEY containing IPSec parameters in MIKEY Init Message.

Step 2- Prepare to send the SIP INVITE using DNS query or resolved host address and send it.

Step 3- Check whether the received provisional response message is from valid User and in the same dialog of SIP INVITE.

Step 4- If the response message is from valid user and same dialog, than

Step 4.1 Process the MIKEY Payloads for authentication, policy check and check the timestamp and copy the RSeq number. In case of mismatch in timestamp, create & send a Cancel SIP message and drop the call.

Step 4.2 Else discard the response and wait for valid response till the timer expires

Step 5- If the 183 Session in Progress response doesn't contain Support/Require Header in SIP, than

Step 5.1- Exit this algorithm and call algorithm (Mikey reply in 200 OK)

Step 5.2- Else, create and send the PRACK message with RAck = RSeq, set with acknowledgement number equals to RSeq received in the 183 Session in Progress provisional response.

Step 6- Generate Session key from the master key using MIKEY.

Step 7- Write the Security Association and policy received in the SA & Policy databases of IPSec by the system calls to kernel space.

Step 8- Resend the PRACK until 200 OK (PRACK) is received till timeout.

Step 9- Process the 200 OK received for PRACK.

Step 10- Process the 180 Ringing Provisional unreliable response message and call the Local ringing tone function.

Step 11- Start the Local Ringing tone.

Step 12- Process the SIP final response to INVITE, 200 OK (INVITE) and stop the Local Ringing tone.

Step 13- Create and send the ACK, acknowledging the Callee about the reception of the 200 OK final response.

Step 14- Create and send the RTP media packets as per security association and policy established.

5.1.2.2 Algorithm for UAS

Step 1- Process the SIP INVITE message received for the identity verification/spam.

Step 2- Process the MIKEY payload for authentication and policy check.

Step 3- If MIKEY authentication fails, than

> Step 3.1- Drop the SIP INVITE message and create & send the 6xx error message.
>
> Step 3.2- Else Process the MIKEY & inspect the IPSec ciphers support at UAS Application and create the MIKEY reply with respect to supported features and policy adopted.

Step 4- If Require header field is present with tag Require= 100req, than

Step 4.1- Create and send 183 Session in Progress response with Require= 100rel and a new field RSeq for sequencing to an initial value indicating Reliable provisional requirement and insert MIKEY Reply in MIME body part with it.

Step 4.2- Else Create and send 183 Session in Progress response without MIKEY reply and call algorithm (mikey response in 200OK) and exit this algorithm.

Step 5- Process the PRACK and check for the RAck value for being in the same dialog.

Step 6- Generate Session Key from the master key received using MIKEY.

Step 7- Write the Security Association and policy received in the SA & Policy databases of IPSec by the system calls to kernel space.

Step 8- Create and send 200 OK for PRACK.

Step 9- Create & send the 180 Ringing provisional message.

Step 10- Make a function call to start the Local Phone Ringing that starts the Phone Ringing.

Step 11- If the User Agent accept/picks the call (OFF the hook), than

Step 11.1- Create and send the SIP response 200OK (INVITE).

Step 11.2- Else create and send BYE after timeout.

Step 12- Process the ACK message, create and send the RTP & RTCP packets.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusions

Call establishment in SIP initiated IPSec suffers from high cryptographic processing at the caller and the recipient end. IPSec is a utility of operating system which requires system calls at kernel space that makes it time consuming. Parameter negotiation for IPSec is done through the SIP INVITE message in calling phase and 200OK response in answering phase. MIKEY reply, sent by responder after picking the phone, contains IPSec parameters which are processed at caller's end that consumes high processing time leading to excessive delay.

The call establishment practices and its delays are studied in this thesis. The high delay of IPSec cryptographic processing at caller end disables the caller to send and receive the RTP media packets at the beginning of the call leading to media transmit clipping and media reception clipping at the start of the call. An Algorithm is provided in which Clipping Effects are eliminated by sending the MIKEY reply in 180 provisional reliable response with PRACK as their acknowledgement, rather than sending MIKEY response in final response 200 OK. This algorithm suffers from ghost ringing, which is the byproduct of above solution, which is eliminated by sending MIKEY reply in 183 provisional reliable response. Hence the problem of Clipping Effects and Ghost Ringing, which arises in IPSec case, has been shifted from answering delay to ringing delay i.e. from answering phase to calling phase.

6.2 Future Scope

A limited number of scenarios have been taken in this thesis. The various other factors, like unsupportable features at the other end etc, can be a source of future study. The few are as below:

- The solution of this thesis leads to increase in the ringing delay from answering delay. Reducing the ringing delay is an area of improvement.
- The PRACK request can be injected by attackers to force retransmissions of reliable provisional responses to cease. As these responses can convey important information, PRACK messages should be authenticated as any other request. NO such authentication of PRACK has been done yet. This PRACK fake request injected by attackers can even led to DOS attack which may include:
    - PRACK flooding
    - Malformed messages (protocol fuzzing)
- This thesis has concentrated on SIP initiated IPSec with Key negotiation & authentication of SIP messages by MIKEY. ZRTP, another media key exchange protocol, can be used for establishing IPSec tunnels directly in RTP media session.

REFERENCES

[1] Douglas E. Comer, "Computer Networks and Internet", Pearson Education Fourth Edition, 2004.

[2] Patrick Park, "Voice over IP Security", Cisco Press, 2009.

[3] Andrew S. Tanenbaum, "Computer Networks", Prentice-Hall India, Fourth Edition, 2005.

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 3550, July 2006.

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", IETF RFC 3261, June 2006

[6] G. Camarillo, Ericsson, "Message Body Handling in the Session Initiation Protocol (SIP)", RFC5621, September 2009.

[7] C. Groves, M. Pantaleo, LM Ericsson, T. Anderson, T. Taylor, Nortel Networks, "Gateway Control Protocol Version 1", IETF RFC 3525, June 2003

[8] T. BernersLee, R. Fielding, Day Software, L. Masinter. "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

[9] Koushik Banerjee, "SIP Tutorial",http://ww.siptutorial.net

[10] J. Rosenberg, H. Schulzrinne, Columbia U, "Reliability of Provisional Responses in the Session Initiation Protocol (SIP)", IETF RFC 3262, June 2002

[11] F. Andreasen, M. Baugher, and D.Wing, "SDP: Session Description Protocol", IETF RFC 4568, July 2004

[12] F. Andreasen, M. Baugher, D. Wing, Cisco Systems, "Session Description Protocol (SDP) Security Descriptions for Media Streams", IETF RFC 4568, July 2006.

[13] S. Kent, R. Atkinson. “Security Architecture for the Internet Protocol”, RFC 2401, Nov 1998.

[14] Ralf Spenneberg, “IPSec HOWTO”, http://www.ipsec-howto.org

[15] D. Harkins, D. Carrel. “The Internet Key Exchange (IKE)”, RFC 2409, Nov 1998.

[16] S. Kent, R. Atkinson. “IP Encapsulating Security Payload (ESP)”, RFC 2406, 1998

[17] S. Kent, R. Atkinson. “IP Authentication Header (AH)”, RFC 2402. Nov 1998.

[18] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, “MIKEY: Multimedia Internet KEYing” IETF RFC 3830, August 2010.

[19] N. Freed, N. Borenstein. “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, IETF RFC 2045, Nov 1996.

[20] N. Freed, N. Borenstein. “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types”, RFC 2046, Nov 1996

[21] B. Ramsdell, “S/MIME Version 3 Message Specification”, IETF RFC 2633, June 1999.

[22] Mohan Krishna, Ranganathan and Liam Kilmartin, “Investigations into the Impact of Key Exchange Mechanisms for Security Protocols in VoIP Networks”,1st IEI/IEE Telecommunication Systems Postgraduate Research Symposium, Dublin, Ireland, 2001

[23] J. Bilien, E. Eliasson, J. Orrblad, and J.O. Vatn, “Secure VoIP: Call establishment and media protection,” 2ndWorkshop Secure VoIP, Washington DC, June 2005.

[24] Andre L. Alexander, Alexander L. Wijesinha, and Ramesh Karne, “An Evaluation of Secure Real-time Transport Protocol (SRTP) Performance for VoIP”, Third International Conference on Network and System Security, 2009.

[25] Liancheng Shan, Ning Jiang, “Research on Security Mechanisms of SIP-based VoIP System”, Ninth International Conference on Hybrid Intelligent Systems, 2009.

[26] Joerg Ott, Elisabetta Carrara, “Extended Secure RTP Profile for RTCP based Feedback(RTP/SAVPF)”,IETF,http://www.ietf.org/internetdrafts/draftietfavtprofilesavpf 01.txt, July 2004.

[27] J. Rosenberg, H. Schulzrinne. “An Offer/Answer Model with the Session Description Protocol (SDP)”, RFC 3264, July 2004.

[28] J. Arkko, F. Lindholm, M. Naslund, K. Norrman, Ericsson, E. Carrara, Royal Institute of Technology, “Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)”, IETF RFC 4567, July 2006.

[29] Si DF, Long Q, Han XH and Zou W, “Security mechanisms for SIP-based multimedia communication infrastructure”, IEEE Conf. on Comm, Circuits and Systems (ICCCAS), ed. Proc. of 2nd ed, pp.575-578, IEEE CS Press, June 2004.

[30] Elhalifa COULIBALY, Lian HAO LIU, “Security Of VoIP Networks”, 2nd International Conference on Computer Engineering and Technology, V3-108(Volume 3), 2010

[31] D.R. Kuhn, T.J. Walsh, and S. Fries, "Security considerations for voice over IP Systems”, Recommendations of the National Institute of Standards and Technology (NIST), Special Publication 800-58, Technology Administration, U.S. Department of Commerce, 2004.

[32] T.J. Walsh and D.R. Kuhn, "Challenges in securing voice over IP," IEEE Security & Privacy, vol. 3, pg 44-49, 2005.

[33] S. Spinsante, E. Gambi, E. Bottegoni, “Security solutions in VoIP applications”, IEEE International Symposium on Consumer Electronics (ISCE 2008), pg 1-4, 2008.

[34] Johan Bilien, Erik Eliasson and JonOlov Vatn, “Call establishment delay for secure VoIP”, WiOpt'04, Cambridge UK, March 2004.

Appendix1: Abbreviations

AH	Authentication Header
AVP	Audio Video Profile
CA	Call Agent
CODEC	Coder-Decoder
DOS	Denial Of Service
DES	Data Encryption Standard
DH	Diffie Hellman
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DNS MX DNS	Mail eXchange record
DNS SRV DNS	resource record for specifying the location of services
CS	Crypto Session
ESP	Encapsulated Security Payload
FW	FireWall
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
IKE	Internet Key Exchange
IETF	Internet Engineering Task Force
ITU	International Telecommunication Union
IP	Internet Protocol
IPSEC	IP Security Protocol
MAC	Message Authentication Code

MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MIKEY	Multimedia Internet KEYing
MIME	Multipurpose Internet Mail Extension
MKI	Master Key Identifier
NAT	Network Address Translation
PCM	Pulse Code Modulation
PKI	Public Key Infrastructure
PSTN	Public Switched Telephone Network
RFC	Request For Comment
RTP	Real Time Transport Protocol
RTCP	Real Time Control Protocol
SA	Security Association
SADB	Security Association Database
SAVP	Secure Audio Video Profile
SDP	Session Description Protocol
SHA	Secure Hashing Algorithm
SIP	Session Initiation Protocol
S/MIME	Secure Multipurpose Internet Mail Extensions
SPD	Security Policy Database
SPI	Security Parameter Index
SRTP	Secure RTP
TCP	Transmission Control Protocol
UA	User Agent

UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UI	User Interface
URI	Universal Resource Identifier
VoIP	Voice over Internet Protocol

List of Publications

Jaspreet Singh, V.P. Singh and Ashish Aggrawal, “Algorithm for elimination of clipping effects & ghost ringing in SIP initiated IPSec VoIP”, International Journal of Computer Application (IJCA), ISSN 0975-8887 (Communicated).