ALTERNATE ENCRYPTION SCHEME FOR REAL-TIME TRAFFIC

A Thesis by

Gopinath Gopalakrishnan

M.S., Wichita State University, 2007

Submitted to the Department of Electrical and Computer Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment for the degree of
Master of Science

July 2007

ALTERNATE ENCRYPTION SCHEME FOR REAL-TIME TRAFFIC

I have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science with a major in Electrical and Computer Engineering.

_____
Ravi Pendse, Committee Chair

We have read this thesis and recommend its acceptance:

_____
Kamesh Namuduri, Member

_____
Krishna K.Krishnan, Member

DEDICATION

This thesis is dedicated to my beloved parents, all my family members and friends who have supported and encouraged me throughout my life. Without their support and encouragement this thesis would not be a success.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Ravi Pendse, for all his support and valuable guidance over the entire course of my academic career at WSU and for giving me an opportunity to do my Master's thesis under his guidance in the field of VOIP Security. I am thankful to my committee for taking the time to work with me in this endeavor.

I would also like to thank Mr. Nagaraja Thanthry, Mr. Amarnath Jasti, Mr. Manivannan Srinivasan and all my friends at the Advanced Networking Research Center (ANRC) at Wichita State University for their help during this time.

.

ABSTRACT

Voice over IP is fast emerging as a strong contender to the traditional circuit-switched PSTN networks. Unlike the PSTN network, which requires dedicated lines, VoIP can share the network that is laid out to carry data traffic as well as other traffic categories. Securing VoIP and other real-time traffic is necessary considering the easy ways of hacking communication over internet. Most of the existing security solutions for VoIP such as IPSec, Secure Real Time Protocol (SRTP) and ZRTP use the standard symmetric encryption algorithms for encrypting voice traffic. In this thesis, the author proposes an alternate encryption scheme that uses PKI architecture for the initial authentication and key exchange, and encrypts the real-time traffic with a symmetric algorithm using a unique key for each packet. The proposed algorithm expected to be less complex compared to the traditional encryption schemes in addition to enhancing the security of the communication. The performance of the proposed algorithm has been simulated by modifying libSRTP. The analysis carried out shows that the proposed scheme helps in improving the voice quality to a certain extent while maintaining the security of the communication.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTD)

CHAPTER **PAGE**

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

INTRODUCTION

Voice communication systems have undergone many changes over the last few decades. Traditional PSTN systems work on the principle of circuit-switching and the data is transported through dedicated lines. PSTN network consists of many dedicated end-to-end circuit connections, which have either analog or digital signaling. Nowadays digital signaling is used more. The major advantages of digital signaling over analog are scalability and reduction of signal corruption.

With the evolution of internet and protocols such as IP, researchers began to focus on convergence of voice and data over a single line. On a packet-switched network such as internet, since there is no dedicated link, the available bandwidth can be utilized for other data. While most of the voice communications are filled with periods of silence, PSTN cannot utilize these gaps for other data since the line is dedicated for that call only. Voice over IP has many advantages over PSTN, it uses internet as a backbone to send voice traffic. It can be combined and transported with other data over the same network. Due to the efficient use of bandwidth the cost of making calls in VOIP is much cheaper compared to PSTN calls. Many organizations are moving from circuit switched PSTN to packet-switched VOIP services. VOIP is not free of challenges, the most prominent of which are quality of service and security. VOIP networks are not nearly as reliable as the PSTN, as they use the same lines to carry all data and voice. Also VOIP traffic is carried over UDP protocol which does not provide any quality of service guarantees. To improve some of these problems, protocols and procedures such as RSVP, queuing policies and traffic shaping were developed.

Security is another major issue in VOIP. Since VOIP is transported over internet, it has the same security issues as standard IP protocols. Some of the problems are eavesdropping, spoofing, denial of service and data corruption. Securing VOIP has been an active area of research and many solutions have been proposed to secure the communication keeping the constraints involved in real-time communication in mind. Since the real-time traffic packets are smaller in size and are processed in real-time they cannot be encrypted like data packets. The most important factor to consider in real-time traffic is limiting the delay of each packet within the maximum acceptable limits. For instance the maximum acceptable delay in case of VOIP traffic is only 150ms per packet. The VOIP traffic is carried over protocols such as IP and RTP. These protocols do not offer much in-built security they have to be secured through other protocols. Securing the real-time packets, accounts to a major part of the packet delay in addition to the network and transport delays. The most common mechanism used for security is encryption of voice packets. Encryption/decryption latency is a problem for any cryptographic protocol, because much of it results from the computation time required by the underlying encryption algorithm. Encryption can be done by either Public-Key cryptography or symmetric encryption. Though Public-Key mechanisms are more secure, the algorithmic complexity involved makes it not suitable for voice application. Also the Public-Key algorithms require a Public-Key Infrastructure (PKI) for their operation which is quite complex.

Most of the existing security protocols such as SRTP and ZRTP use symmetric algorithms for encrypting the real-time traffic. All these algorithms share a key on a per session basis and encrypt the digitized voice. There is also research going on trying to implement PKI architecture for VOIP and other real-time traffic. This thesis compares the features of the

existing security protocols such as SRTP, ZRTP and IPSec for Real-time traffic. Also an alternate encryption scheme that encrypts the payload on a per packet basis using a unique key for each packet is proposed. The proposed scheme uses PKI architecture for the initial authentication and key exchange mechanism and uses symmetric encryption algorithm for encrypting the actual payload. The proposed scheme provides high security by using a unique key for each packet and by the PKI architecture for call setup. The delay during call is reduced by using a less computationally intensive encryption algorithm that causes less packet delay. The rest of the chapter gives a brief overview of RTP. Chapter 2 discusses the features of existing security protocols such as SRTP, ZRTP and IPSec. Chapter 3 discusses the types of encryption and their features. Chapter 4 will outline the features of the proposed Alternate Encryption Scheme and the protocol procedure. Chapter 5 will describe how the simulations and tests were done and their results and also a brief discussion of the pros and cons of the proposed scheme. And finally Chapter 6 will discuss the future work that could be done.

CHAPTER 2

EXISTING VOIP SECURITY PROTOCOLS

In the Chapter 2 the following section provides a brief overview of RTP protocol. The further sections discusses about the features and working procedure of existing VOIP security protocol such as SRTP, ZRTP and IPSec. Also the possible drawbacks of the protocols in securing VOIP are analyzed.

2.1 RTP (Real-time transport protocol)

Real-time transport protocol defines a framework to carry real-time traffic such as voice and video over the internet. The RTP protocol does not have a standard transport layer protocol it can work with both TCP and UDP. Due to usage with real-time traffic RTP is generally used with UDP protocol. RTP does not provide any in-built security feature and does not guarantee any Quality-of-service. It relies on the other layers to provide them. It supports both unicast and multicast applications. RTP operates in conjunction with RTCP which is the control protocol for the RTP traffic. RTCP monitors the quality of service and control information of the participating users.

The congestion and flow control for the RTP and RTCP packets are achieved by the transport layer protocols. In the RTP transport the audio and video sessions are transported and received as different media sessions. The advantage of this method is that the lower bandwidth data doesn't have to wait till the higher bandwidth data is transported.

The packet structure of RTP is as shown below,

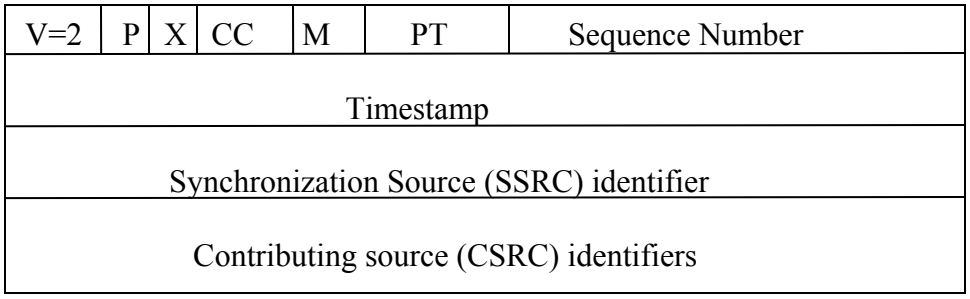


Figure 1: RTP Packet format [1]

Version (V) – The Version field denotes the version of RTP. The current version number is two.

Padding (P) - If the padding field is set the payload contains additional padding data. Padding is required by some of the cryptographic algorithms to maintain block sizes.

Sequence Number - It is used to detect packet loss and to re-order out of sequence packets at the receiver side.

Timestamp - The timestamp denotes the sampling instant of the first octet in the RTP packet. The sampling instant is derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations [1].

Synchronization source identifier (SSRC) - The SSRC field is used to identify the synchronization source of the RTP session. It is generally chosen in a way that two sources of a same RTP session do not have the same SSRC identifier.

Contributing source identifier (CSRC) – The CSRC identifier contains the entire contributing sources list for the payload of a particular RTP packet.

2.2 SRTP (Secure Real-time Protocol)

SRTP is a security profile for RTP that adds confidentiality, message authentication, and replay protection to that protocol [2]. It provides a framework for encryption and authentication of RTP and RTCP traffic. SRTP defines a set of default cryptographic transforms to be used and also provides room for new transforms to be introduced. The Basic functionality of SRTP is to ensure confidentiality for data and the control messages by encrypting the respective payloads. Also to provide security mechanisms for both unicast and multicast RTP applications and protection against replayed packets. SRTP framework allows for upgrading with new cryptographic transforms. It has a low bandwidth and computational cost. SRTP provides limited packet expansion and high tolerance to packet loss and reordering. SRTP is a bump-in-the-stack implementation; it fits into the RTP easily. The default ciphers used in SRTP are fast stream ciphers (the default is AES counter mode).It also provides fast message authentication.

Features and framework of SRTP

- A single “Master Key” provides keying material for RTP and corresponding RTCP streams. The session keys used for encryption are derived from the master key through a cryptographically secure way.
- A “Salt Key” which can be made public is used along with the master key to protect against pre-computation attacks.
- Synchronization of the packets and protection against packet loss is accomplished using sequence numbers. Replay attacks are tackled by comparing the incoming packet with a replay list maintained by the receiver.

- SRTP is a bump-in-stack implementation that resides between the RTP profile and transport layer.
- SRTP does not define any particular key exchange mechanism, the most commonly used mechanisms are MIKEY and IKE.

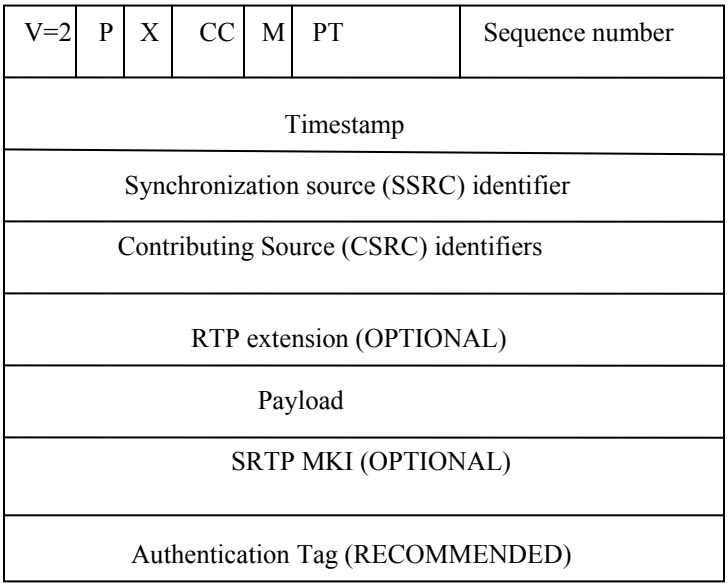


Figure 2: SRTP Packet Format [2]

- The sequence number is a 16 bit number extracted from the RTP packet header
- SSRC (Synchronization source identifier) helps to synchronize the SRTP packets. SRTP streams within a given RTP session are identified using their SSRC. The streams may share a single master key and session keys and separate replay lists and packet counters are maintained for each SSRCs
- The RTP padding is used by cryptographic transforms that require padding. The default transforms defined in current SRTP do not require any padding.

- MKI (Master Key Index) - The MKI identifies the master key from which the corresponding session keys were derived. The MKI is generally used for identifying a particular master key within a cryptographic context and for key management purposes.
- Authentication Tag - used to carry authentication data. It provides authentication of the RTP payload and header information.
- Rollover counter - is a 32 bit unsigned counter that records the number of times the 16-bit SEQ number has been reset to zero after passing through 65,535.
- Packet Index - The packet index of the SRTP packet is calculated as,

  Packet Index, I = 2^16 * ROC + SEQ.
- Replay list – A replay list is maintained on the receiver side using which an incoming packet is compared to check for previously received packets and replay protection is provided.

In addition to these parameters SRTP consists of a "Master Key" which has to kept secret and a "Salt Key" which can be made public. Also there are cryptographic transform dependent parameters such as IVs (Initialization vector), block size of ciphers , session keys etc.

2.2.1 SRTP Packet Processing

Sender Side [2]

1. The cryptographic context is determined from the SSRC of the stream. The cryptographic context is identified by the context id,

   context id = <SSRC, destination network address, destination transport port number>
2. The packet index is determined from the rollover counter and sequence number. Packet index, i = 2^16 * ROC + SEQ.
3. From the packet index the corresponding "Master Key" and "Salt Key" are determined.

4. The session keys to be used for encryption are derived from the master key and salt key using the key derivation algorithm.
5. Using the encryption algorithm defined in the cryptographic context and the session encryption keys derived above the RTP payload is encrypted.
6. Using the authentication mechanism defined in the cryptographic context the authentication tag is added to the packet.
7. The ROC is updated and the packet is sent to the receiver side.

Receiver Side [2]

1. The cryptographic context used is determined from the SSRC.
2. The packet index is calculated using the rollover counter and the sequence number,

   The receiver estimates the index as $I = 2^{16} * v + SEQ$,

   Where v is chosen from the set {ROC-1, ROC, ROC+1} (modulo 2^{32})

   Such that I is closest (in modulo 2^{48} sense) to the value $2^{16} * ROC + s\_l$
3. The session keys are derived for the corresponding "Master Key" and "Salt Key".
4. The packet is checked for replay by comparing it with the received packets in the replay list.
5. Using the decryption algorithm defined in the cryptographic context the encrypted portion is decrypted.
6. The ROC is updated and the authentication tag is removed.

2.2.2 Replay Protection

A packet is said to be replayed when it has been stored by a hacker or attacker and reinserted into the traffic. Such attacks are countered by SRTP using replay protection. The SRTP receiver maintains a replay list that contains the indices of all the packets received. The

incoming packet index is checked with this replay list to determine if it has already been received. Sliding window mechanism is used for providing replay protection. Only packets with index greater than the window are packets within window not already received are accepted by the receiver.

2.2.3 Encryption Process in SRTP

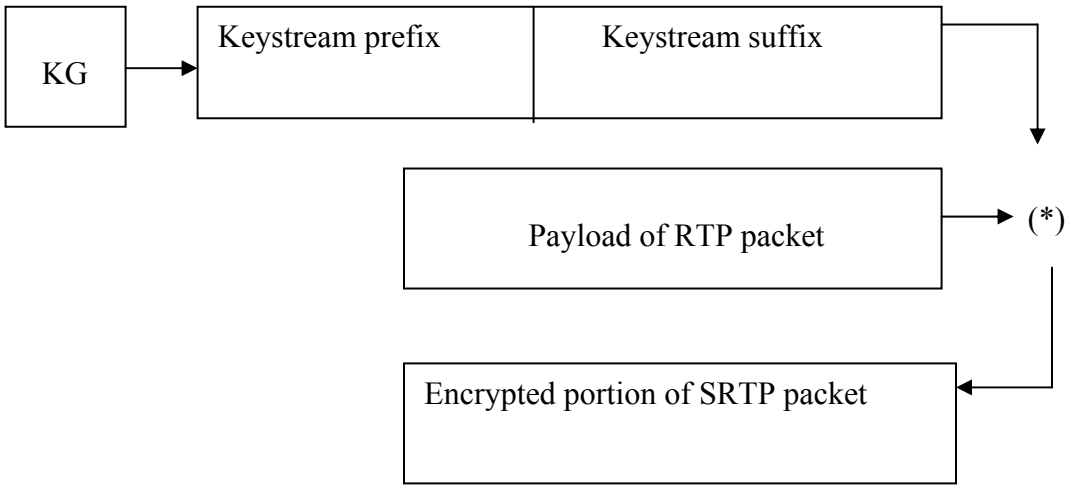


Figure 3: Default SRTP Encryption Processing [2]

Here KG denotes keystream generator, and (*) denotes bitwise exclusive-or.

From the "Master Key", "Salt Key" and the packet index, the corresponding "Session Key" for the packet is derived using the key derivation algorithm. The session key is then XOR ed to the payload of the RTP packet to produce the corresponding SRTP packet. When the bit-size of the keystream generated is greater than the payload the excess keystream (least significant portion) is discarded. The generation of keystream depends on the type of cipher defined in the particular cryptographic context. The default ciphers used in SRTP are AES operating in counter mode and F-8 mode.

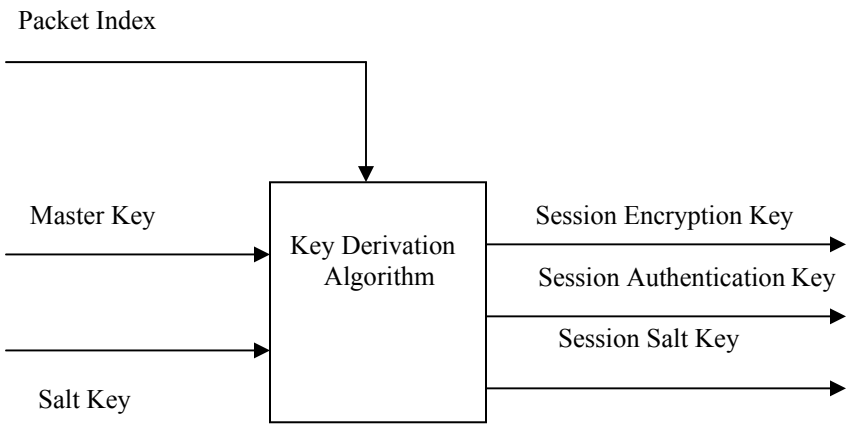


Figure 4: SRTP key derivation [2]

2.2.4 Authentication mechanism

The default defined authentication mechanism for SRTP is HMAC-SHA1. The authentication tag is computed by the sender and appended to the data. The receiver side computes the authentication tag using the algorithm defined in the cryptographic context and compares it to the tag of the received message. The data is authentic if both the tags are valid otherwise it is invalid.

2.3 SRTCP (Secure Real-time Transport Control Protocol)

SRTCP is the control protocol for SRTP. Three new fields SRTCP index, E flag and Authentication tags are added to the RTCP packet. E- Flag is a single bit value that indicates whether the SRTCP packet is encrypted or unencrypted. Value of "1" indicates encrypted packet and "0" indicates unencrypted packet.

SRTCP index – is a 31 bit counter for each SRTCP packet. It is explicitly defined in each

packet compared to SRTP where the index is implicit.

Authentication TAG – it carries the authentication value of the message.

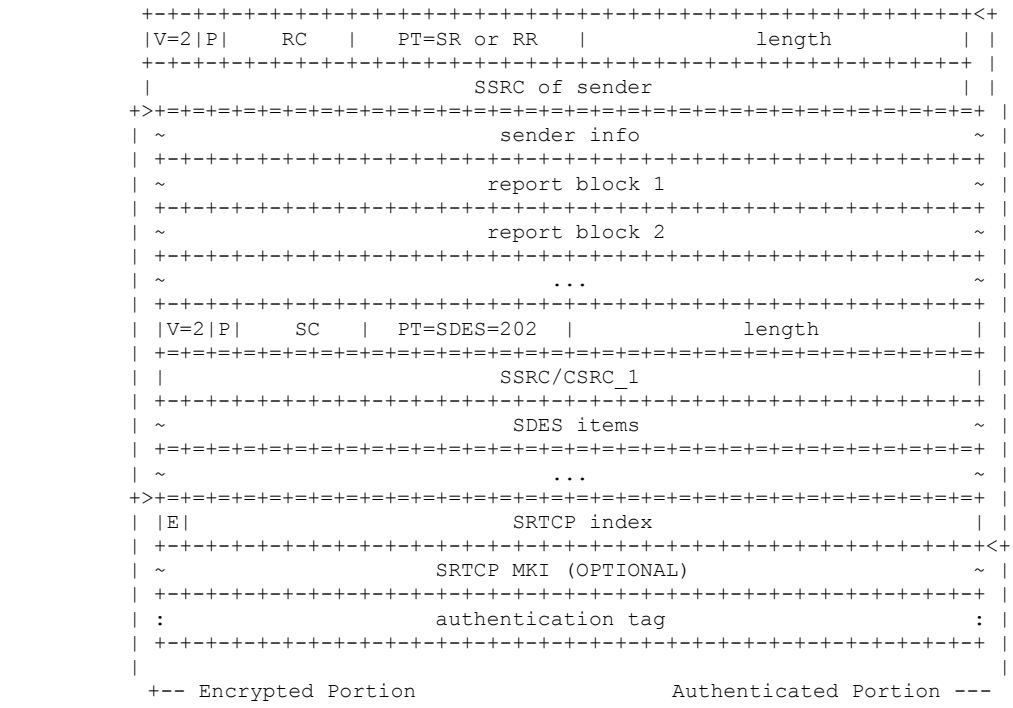


Figure 5: SRTCP Packet Format [2]

2.4 ZRTP

ZRTP is an extension to RTP and defines a key exchange protocol for the SRTP based on the Diffie-Hellman mechanism. ZRTP was invented by Phil Zimmermann and his team. It borrows some of the ideas from PGP based PGPfone, which was the developer's earlier product. The key exchange is done on a peer to peer basis through the ephemeral Diffie-Hellman mechanism. During the call setup phase a shared secret is generated by the communicating peers using the Diffie-Hellman mechanism, which is then used to generate keys and salt for the SRTP session [3]. Even though Diffie-Hellman is based on Public-Key algorithm ZRTP does not require Public-Key infrastructure (PKI). The keys used during a session are destroyed after the call to achieve perfect forward secrecy. But some of the keying material sent in a session is cached and mixed with the next call's shared secret.

This provides continuity property to the key stream analogous to the SSH [3]. The possibility of man-in-the-middle attack (MiTM) in Diffie-Hellman mechanism is countered by SAS (Short authentication mechanism). In SAS an authentication string is displayed to the peers to be read aloud and compared before the start of the actual key exchange. Protection against MITM is also provided by the key continuity process. ZRTP is independent of the signaling layer and is compatible with any of the signaling protocol like SIP, H.323 and Peer- to- Peer SIP [4]

2.4.1 Overview of ZRTP working

The working of ZRTP process can be summarized in three basic steps,

1. The initial signaling process and detection of the peer's capability to support ZRTP.
2. The second step is the actual key exchange mechanism using the DH protocol, and in the final phases the protocol transitions to the SRTP mode.

ZRTP process starts after the initial signaling process is done using signaling protocols such as SIP. One of the two communicating peers initiates by sending a ZRTP Hello message to the other endpoint [3]. The Hello message determines if the other end point supports ZRTP, also determines the SRTP configuration options and ZID common between the two peers. ZID is a unique 96 bit random ZRTP ID generated before every session. It is used to find previous shared secrets if there are any with the same peer. If the other endpoint supports ZRTP and agrees upon the parameters it sends reply in the form of HelloACK message. Once the Hello ACK is received the initiator sends the commit message and the Diffie-Hellman Public-Key values are exchanged. And the SRTP master and salt keys are determined from the shared secret.

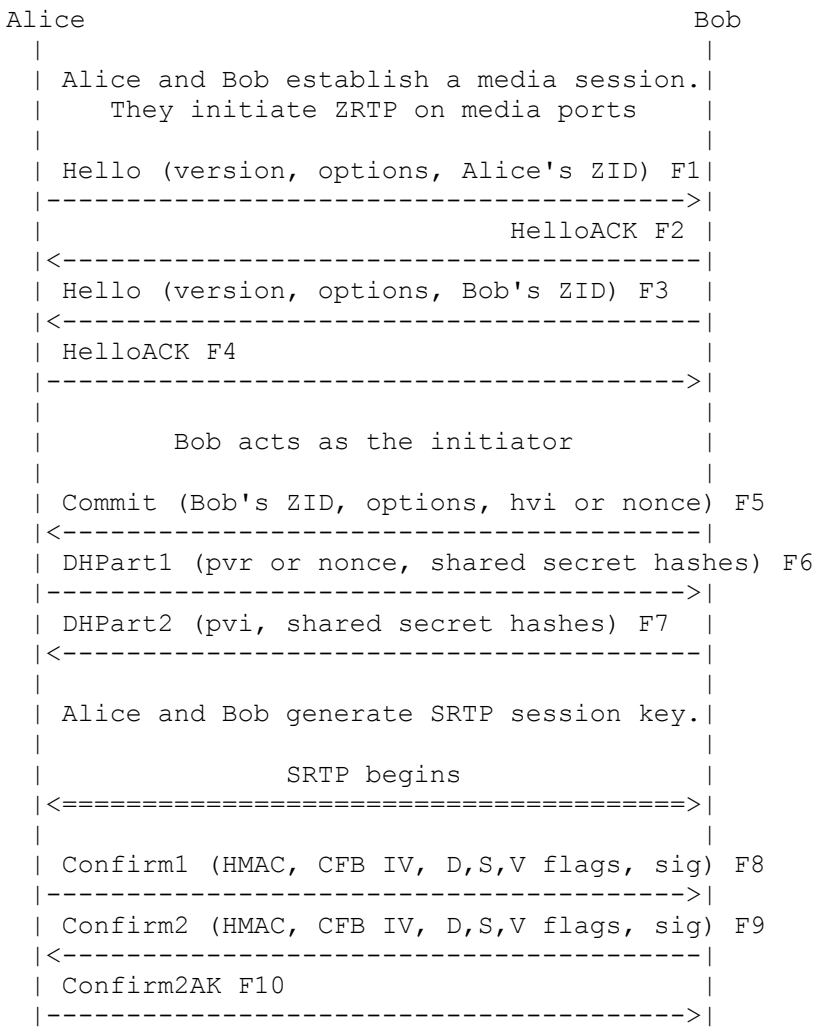


Figure 6 : ZRTP Call Setup Process [3]

The key exchange can take place either in pre-shared mode or Diffie-Hellman mode. In the preshared mode the Diffie-Hellman calculation can be omitted if the end points have shared secrets from the previous session. Further details about the key exchange and shared secret calculation mechanism can be found in the ZRTP RFC [3].Once the shared secret is determined the SRTP master and salt keys are derived from it for each direction. ZRTP supports AES-CM encryption mode with either 128 or 256 bit key length. HMAC1 32 bit and 80 bits are the supported authentication algorithms.

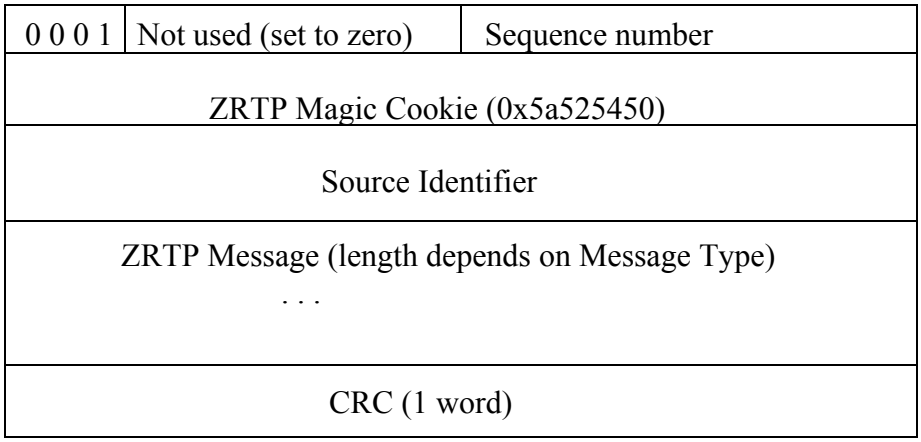


Figure 7: ZRTP Packet Format [3]

ZRTP magic cookie is used to uniquely identify a ZRTP packet and has a value of 0x5a525450 [3]. Source identifier is the SSRC of the corresponding RTP stream.

A 32 bit CRC checksum is defined in each packet to detect transmission errors [3].

2.4.2 Short Authentication String (SAS)

Before the start of key exchange mechanism the communicating peers read aloud an authentication string to each other to counter MITM. The SAS type is decided by the users during the commit message. When the SAS is verified successfully the SAS verified flag is set. If the SAS do not match, it implies that there is MITM. The cached key continuity also helps in defending against MITM attacks. The SAS MAY be signed and the signature sent using the Confirm1 or Confirm2 messages. The signature algorithm is also sent in the Confirm1 or Confirm2 message, along with the length of the signature.

The signature exchanged in the encrypted Confirm1 or Confirm2 messages MAY be used to authenticate the ZRTP exchange [3].

2.4.3 Security issues in ZRTP

The ZRTP protocol doesn't use any PKI architecture even though it is based upon Diffie-Hellman Public-Key exchange mechanism. The major security issue in Diffie-Hellman is the man-in-the-middle (MITM) attacks. Protection against such attacks is generally accomplished by deploying a well managed PKI. PKI architecture requires centrally managed CAs (Certification Authority) and RAs (Registration Authority). There has to be network between all the CAs and a mechanism for certification flow. Maintaining a PKI maybe a tedious activity, ZRTP instead used SAS mechanism to protect from the MITM attacks. The issue with the SAS mechanism is that inattentive users who skip the SAS or unattended calls to answering machines may pave way for MITM.

There are also threats due to voice spoofing attacks where an impersonator can spoof the identity of the real user even before the call is initiated. There are also issues with how SAS can be implemented in end equipments where a display or GUI is not available.

2.5 IPSec in VOIP

IPSec is a group of protocols that operate together to secure the Internet protocol (IP). IPSec provides encryption, authentication and replay protection to the IP packets. It is a framework for security that operates at the network layer. It has the flexibility to secure all the network layer protocols such as TCP, UDP and other higher layer protocols.

IPSec consists of three separate protocols,

- Authentication Header (AH): provides authenticity guarantee for packets, by attaching strong crypto checksum to packets, it verifies that:
    1. The packet was originated by the expected peer. The packet was not generated by impersonator.

2. The packet was not modified in transit.

- Encapsulating Security Payload (ESP): provides confidentiality guarantee for packets, by encrypting packets with encryption algorithms. It provides the required security services, such as data origin authentication, confidentiality, integrity, protection against replay and limited traffic flow confidentiality.
- Internet Key Exchange (IKE): IKE is the key negotiation protocol used by IPSec for negotiating the encryption methods, authentication methods and the keys for them. The IKE key negotiations are done through the IKE security associations (SA).

IPSec has two modes of operation – Tunnel mode and Transport mode.

Tunnel Mode – In tunnel mode the whole IP packet is encrypted and encapsulated into a new IP packet. Tunnel mode is used for network-to-network communications (secure tunnels between routers) or host-to-network and host-to-host communications over the internet [5].

Transport Mode - In transport mode only the payload part of the IP packet is encrypted. In transport mode no changes is made to the IP header. It is the default mode of IPSec operation.

2.5.1 Security Associations (SA)

Security association can be defined as an agreement between two communicating parties. It defines and negotiates the security services to be used between the two parties. Separate SAs are setup for encryption, authentication and key exchange for a particular communication session between two parties.

The SAs are one way that is for a bi-directional communication two SAs are negotiated. The SA has fields like destination address, keys and transforms used for the session and a security parameter index (SPI). The SPI number is mentioned in the AH and ESP packets to

indicate the SA used for the particular packet. The SAs are generally stored in a database called Security Association Database (SAD). SAs can also be encapsulated within SAs to form bundles. For instance one SA may protect communication till the gateway and another SA can protect the communication to a host. The SAs can be encapsulated and routed as SA bundles [6]. The first SA that is negotiated for a communication session is the IKE SA. Then a pair of SAs are negotiated for the AH and ESP protocols. A security policy database is maintained by IPSec that defines what action has to be taken for each packet.

2.5.2 Problems with IPSec for VOIP

In transport mode only the data is encrypted and the IP header and IPSec header or left unprotected. In tunnel mode, both the payload and the headers are encrypted. The encryption and decryption processes in ESP result in increased latency and ESP provides no authentication, there the two protocols have to be used together for securing VOIP traffic which may result in increased latency. The size of the IP packet also increases with the IPSec header; this increases the latency and jitter in the VOIP packets. The increased size also reduces the throughput of the crypto engine. The most important issue with IPSec for VOIP traffic is the incompatibility between IPSec and NAT. NAT traversal completely invalidates the purpose of AH because the source address of the machine behind NAT is masked from outside world, therefore the true sender of the data cannot be authenticated [7].

CHAPTER 3

PUBLIC-KEY AND PRIVATE KEY ENCRYPTION

The principle and working procedure of Public-Key and Private Key encryption is described in the following sections. The section 3.1.1 describes the working of Public Key Infrastructure (PKI). Also the performance of various Symmetric and Asymmetric algorithms is analyzed for securing voice traffic.

3.1 Public-Key Encryption

Public-Key encryption, also known as asymmetric encryption uses a pair of keys for securing the data. This method allows users to communicate securely without having prior access to a secret key. The Public-Key as the name implies can be made public and is generally used for encryption whereas the private key is kept secret and is used for decryption. The two keys are related mathematically the Public-Key is generally derived from the private key. The secrecy of the Public-Key encryption lies in the fact that it is impossible to derive the private key from the Public-Key. One of the most important issues with Public-Key encryption is effective key management and proving that the Public-Keys are authentic. This has to be done in order to avoid the man-in-middle attacks. It is the type of attack where a hacker resides in between two communicating parties and traps their communication by impersonating to be the genuine party. For example consider a communication between two parties Alice and Bob, Alice first gets Bob's Public-Key from Bob and encrypts the data with that and sends it. When a malicious attacker Eve is able to impersonate and send her Public-Key as Bob's key man-in-the middle attack takes place. Now Eve can act as Bob to Alice and as Alice to Bob and trap the whole conversation without their knowledge. The solution to this issue is authentication of the two

parties before communication. This is achieved by having a Public-Key infrastructure where the authenticity of the keys is verified by third parties generally known as Certificate Authorities

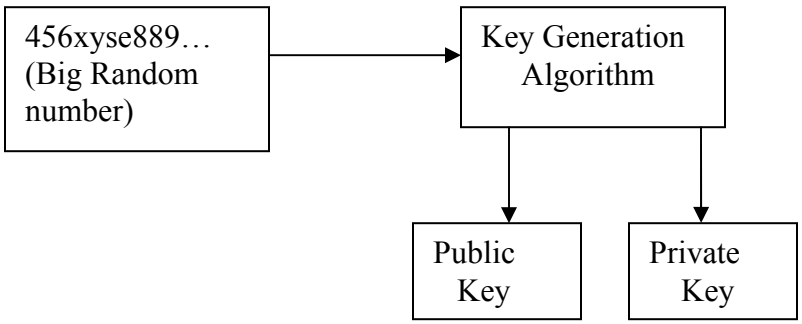


Figure 8 Public-Key Cryptography Key Generation

The Public-Key encryption algorithms are generally slower than the symmetric encryption schemes. For this reason the Public-Key encryption algorithms are used for key exchange mechanisms and symmetric algorithms are used for encryption purposes.

3.1.1 Public-Key Infrastructure (PKI)

Public-Key Infrastructure (PKI) is a mechanism to provide a trusted network to manage key distribution and authentication for Public-Key cryptography. The confidentiality, integrity and authenticity of the communicating parties can be verified through PKI. Authentication of the keys and key management is implemented through third party called the certificate authority (CA). The two main components of a PKI are certificate authority (CA) and registration authority (RA). The registration authority (RA) is the one that verifies the authenticity of the user or the company requesting for the certificate. The RA verifies the user/company's possession of the private key and the corresponding Public-Key [8].

After verifying the identity the request is forwarded to the CA. Generally the CA issues, verifies, stores and revokes the certificates of users. The certificate contains information such as Public-Key of the user, validity period, name of the subject and name of the CA who issued the certificate. X.509 is the most widely used standard for the digital certificates. When Alice and Bob communicate through PKI, Alice sends a request message to Bob encrypted with her private key. Bob then decrypts the message with Alice's Public-Key and sends a accept message back to her encrypted with his private key. The two parties also verify each other's certificate and its validity. After the initial authentication they send messages encrypted with the other party's Public-Key which can be decrypted only by the intended recipient.

3.1.2 Working of PKI

Consider two parties Alice and Bob who want to communicate with each other through the PKI, the steps involved in setting up the communication session are as follows,

- Lets say Alice is the initiator , firstly she applies a certificate with the RA
- The RA verifies the identity of Alice and forwards it to the CA if found to be genuine
- The CA verifies the authenticity of RA and examines the certificate request and is found acceptable issues a certificate. The certificate is also stored in a certificate repository.
- Alice then sends a message to Bob encrypting with his Public-Key. The certificate is also included in the message.
- Bob retrieves the certificate and checks its validity and authenticity. Bob then sends a response message encrypting it with Alice's Public-Key.

The issues in PKI are the number of CAs required for a communication session. And who will verify the authenticity of the CAs and RAs. Also how are the certificates stored and revoked. There are many trust models in PKI such as Top-down trust model, Up-cross-down model and

Bridge PKI. Research work has been done by Chen Jie to find the best suited model for VOIP architecture [8].

3.1.3 Diffie-Hellman Key exchange mechanism

Diffie-Hellman key exchange mechanism based upon Public-Key cryptography was developed by Diffie and Hellman in the year 1976. It is a protocol that enables two users without any prior association between them to derive a common shared secret key. This shared key can then be used in symmetric ciphers to encrypt data. For instance, if Alice and Bob want to generate a shared secret based on the Diffie-Hellman key exchange mechanism. First, Alice generates a random private value *a* and Bob generates a random private value *b*. Both *a* and *b* are drawn from the set of integers. Then they derive their public values using parameters *p* and *g* and their private values. Alice's calculates her public value as $g^a \bmod p$ and Bob's calculates his public value as $g^b \bmod p$. They then exchange their public values. Finally, Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$. Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key *k*.[RSA lab link]. The values *p* and *g* are publicly available to all users, *p* is a prime number.

Diffie-Hellman key exchange can be done in two different modes – ephemeral static mode and static-static mode. ZRTP uses ephemeral Diffie-Hellman key exchange mechanism to derive the master and salt keys for the corresponding SRTP session. In the ephemeral static mode the sender has to generate a new key pair for each message and the recipient has a static certified key pair [3]. In the static-static mode both the sender and recipient have certified static keys.

3.1.4 Issues with Diffie-Hellman Key exchange

The most important issue with Diffie-Hellman key exchange mechanism is that it lacks an in built authentication mechanism. Due to this the Diffie-Hellman mechanism is susceptible to man-in-the- middle (MITM) attacks. Consider a communication session between Alice and Bob, is a hacker Eve is able to intercept the messages from Alice to Bob and can act as Alice to Bob and Bob to Alice MITM attacks take place. In this case the hacker in the middle can modify the original messages or be a passive listener to the communication session. The other issue with the Diffie-Hellman key exchange or any other Public-Key mechanism is the distribution and storage of the keys and certificates.

3.1.5 RSA

The RSA encryption algorithm was developed in 1977 and is named after its inventors Ronald L.Rivest, Adi Shamir and Len Adleman. The principle behind RSA is based on the difficulty in factoring large numbers. It is used both as a full fledged Public-Key encryption algorithm as well as an authentication algorithm for digital signatures. The security of the RSA algorithm is based on finding large prime numbers. They should be large enough and random. The key generation mechanism of RSA is as explained below,

- Two large primes p and q are generated and their product n = pq is calculated.
- Value of phi is calculated as (φ) phi = (p-1)(q-1)
- Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1
- The secret exponent d, 1<d<phi is calculated such that ed = 1(mod phi)
- The Public-Key value is determined as (n,e) and the private key value is (n,d) [19]

Here the value is n is generally known as modulus. The values of p, q and phi are secret [19]. The cipher text is calculated as $C = m^{\wedge e} \bmod n$. Where (n,e) is the Public-Key of the receiver.

For decryption of the cipher text the receiver calculates the value of $m = c^{\wedge d} \bmod n$. Where (n,d) is his private key. When used RSA must be combined with some form of padding scheme, so that no values of *M* result in insecure cipher texts. RSA used without padding may suffer from a number of potential problems [10].

3.1.6 Issues with RSA

The security provided by RSA is based upon the difficulty in factoring large numbers and difficulty in solving the RSA problem. The RSA problem is defined as the task of taking *e*th roots modulo a composite *n*: recovering a value *m* such that $c=m^e \bmod n$, where (*e*, *n*) is an RSA Public-Key and *c* is an RSA cipher text [10]. Like Diffie-Hellman mechanism RSA algorithm is also susceptible to man-in-the-middle attacks. Such types of attacks are countered using PKI and other means. RSA like other Public-Key algorithms is pretty slower than symmetric algorithms therefore it is generally preferred only for authentication mechanism in VOIP applications. There are also a few attacks such as timing attacks and adaptive chosen cipher text attacks that have to be guarded against.

3.2 Symmetric Key Encryption

Encryption algorithms that use a single shared key or trivially related keys for both encryption and decryption are called Symmetric Key encryption algorithms. Symmetric key algorithms are generally much faster than Public-Key algorithms. The security provided by symmetric key algorithms depends mainly on the secure sharing of the symmetric key between the communicating parties. Symmetric key algorithms do not require any maintenance intensive PKI architecture. The following figure explains the basic principle of encryption and decryption using symmetric key algorithms.

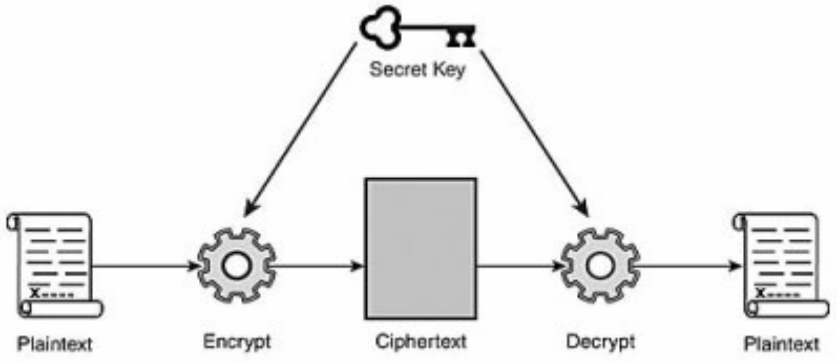


Figure 9 : Encryption and Decryption Process [20]

Symmetric key algorithms can generally be classified as Stream and Block ciphers.

- Stream Cipher – In stream ciphers the encryption is done on a bit-by-bit basis. Encryption is done by combining the keystream with the plaintext, usually with the bitwise XOR operation [9]. Stream cipher can be classified as synchronous and self-synchronizing stream ciphers. In Synchronous stream cipher the pseudorandom keys are generated independently of the plain text and cipher text. The sender and receiver must be exactly in step for decryption to be successful. If digits are added or removed from the message during transmission, synchronization is lost [11]. The self-synchronizing stream cipher uses the some of the previous N cipher text digits to calculate the keystream. The receiver will automatically synchronize with the keystream generator after receiving *N* cipher text digits, making it easier to recover if digits are dropped or added to the message stream [11].
- Block Cipher - In Block ciphers the encryption is done on a group of bits or a block by block basis. Generally block sizes of 64 or 128 is used. Mostly encrypting the same plaintext with same key produces same ciphertext in block ciphers. Block cipher can operate in different modes like CBC, ECB, OFB and CFB. All these modes except ECB require an IV (initialization vector) which provides randomization of the process [12].

This section describes and compares some of the well known symmetric encryption algorithms.

3.2.1 Data Encryption Standard (DES)

DES is a symmetric block cipher algorithm developed in the 1970s. The DES algorithm operates in block sizes of 64 bits and uses a key of 64 bits. Out of the 64 bits of the key 8 bits are used for parity checking therefore the effective key length is 56 bits only. Generally the same key material is used for both the encryption and decryption processes. The DES algorithm consists of sixteen iterations or rounds. They also have initial and final permutations. The basic design consists of eight substitution boxes. DES can operate in ECB, CFB, CBC and OFB modes. The DES algorithm is now considered insecure as the 56 bit key does not provide enough security considering the latest computational advances. DES keys can now be broken within hours. DES has been succeeded by Triple-DES and AES.

3.2.2 Triple DES

Triple DES was developed as a successor to DES when the latter was found to be providing less security. The Triple DES could be considered as just another mode of operation of the DES algorithm. The key length used by Triple DES is 192 bits and it operates as a block cipher. The encryption process is similar to DES it is just repeated three times in Triple DES, therefore the name Triple DES. The payload is encrypted with the first key then it is decrypted with the second 64 bit key and finally encrypted with the third 64 bit key. The decryption process is just the reverse of the encryption process.

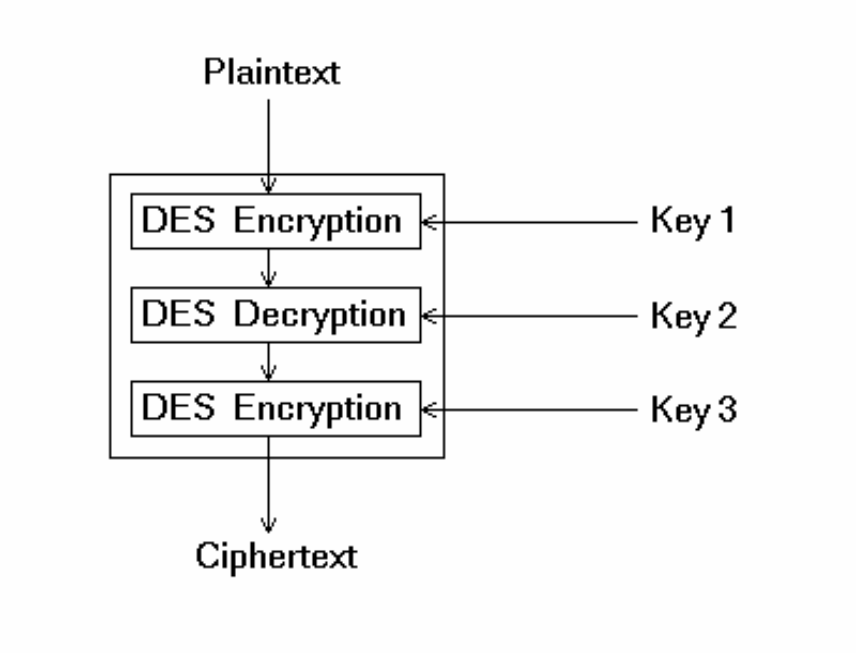


Figure 10 Triple DES [13]

Triple DES requires roughly three times more time and computational power than DES algorithm. Triple DES can operate in Triple ECB and Triple CBC modes. The Triple ECB mode works in the same way as the DES ECB mode. The triple CBC mode acts in a similar way as DES CBC mode it adds an extra layer of security to Triple DES and is therefore more secure than Triple ECB, although it is not used as widely as Triple ECB [13]. Due to its design the Triple DES is slower when deployed in software and latest processors. It is generally more preferred in hardware implementations.

Triple DES is generally more used for data encryption rather than voice encryptions due to its large computational time. It is being replaced by its successor AES in most of the applications. The next section presents more details on AES and its operation.

3.2.3 Advanced Encryption Standard (AES)

AES also known as Rijndael is a block cipher developed by Belgian cryptographers Joan Daeman and Vincent Rijmen. The key sizes used in AES are 128, 192 and 256 bits.

AES design is based on substitution-permutation network. AES operates faster than Triple DES both in hardware and software implementations. The encryption process of AES consists of four stages – AddRoundKey, SubBytes, ShiftRows and MixColumns. The last three rounds are based on confusion and diffusion techniques.

- AddRoundKey - in this stage each byte is combined with a round key. The round key is derived from the key schedule [14]
- SubBytes - it is the step where the bytes are substituted with values from a lookup table.
- ShiftRows - each row of the state is cyclically shifted for a certain number of steps [14].
- MixColumns - a mixing operation which operates on the columns of the state, combining the four bytes in each column using a linear transformation [14]

Currently most of the data and voice applications use AES algorithm for security. SRTP uses AES counter mode and F8 mode as the default encryption algorithms. The NIST has defined five modes of operation for the block ciphers – ECB, CBC, CFB, OFB and CTR.

AES CTR mode

The AES counter mode used in SRTP and ZRTP works on the principle of encrypting successive integers [2]. In this mode each packet is encrypted with a different keystream. The unique each packet value is generated using the initialization vector (IV). The keystream can be pre-computed as it does not depend on the payload of the packet and is calculated from the IV. The AES counter operates on block size of 128 bits and key sizes of 128, 192 and 256 bits. The AES CTR mode operates with higher speed compared to other algorithms. AES-CTR uses the only AES encrypt operation (for both encryption and decryption), making AES-CTR implementations smaller than implementations of many other AES modes [15].

AES F8 Mode

The AES F8 mode used in SRTP is a variation of the output feedback mode (OFB). It has been designed with a more elaborate initialization and feedback function [2]. The key sizes used are same as that of AES CTR mode. This mode is also used in the UMTS 3G mobile networks.

	DES	**3DES**	**AES**
Key Size	56 bit key	192 bit key	128 bit key
Complexity	Less complex and requires less computational power	Very complex And requires more computational power and time	Moderately complex and requires less computational time and power compared to 3DES
Security	Least secure	More secure with 192 bit key	Comparable to 3DES
Operation Mode	Block cipher	Block Cipher	Block Cipher

Table 1 Comparison of Symmetric Cryptographic Algorithms

CHAPTER 4

ALTERNATE ENCRYPTION SCHEME

Since the real-time traffic packets are smaller in size and are processed in real-time they cannot be encrypted like data packets. The two major concerns during real-time communication are, maintaining the delay of each packet within the maximum acceptable limits and providing high level of security. The maximum acceptable delay in case of VOIP traffic is 150ms per packet. The encryption mechanisms employed in VOIP account for major part of this delay. Encryption/Decryption latency is a problem for any cryptographic protocol because, much of it results from the computation time required by the underlying encryption algorithm. Encryption can be done by either Public-Key Cryptography or Symmetric encryption as described in the previous sections. Generally, symmetric encryption algorithms are preferred over Public-Key encryption in VOIP applications as they are comparatively faster. Symmetric key algorithms employ the same key for both encryption and decryption and hence their security is solely dependent on the secure handling of the symmetric key. This section proposes an alternate encryption scheme for encrypting real-time traffic that uses symmetric algorithm for encrypting the packets and PKI architecture for authentication and key exchange. Some of the ideas have been borrowed from SRTP and ZRTP mechanisms.

4.1 Features of the Scheme

- Each packet is encrypted with a unique key using a symmetric encryption algorithm.
- The initial key exchange and authentication is accomplished using PKI architecture.
- The initial keys are exchanged during the call setup and the subsequent keys are exchanged during the call session.

- A unique key is generated and used for each packet, and hence session key attacks are not possible. The security of any session is not based upon the security of a particular key.

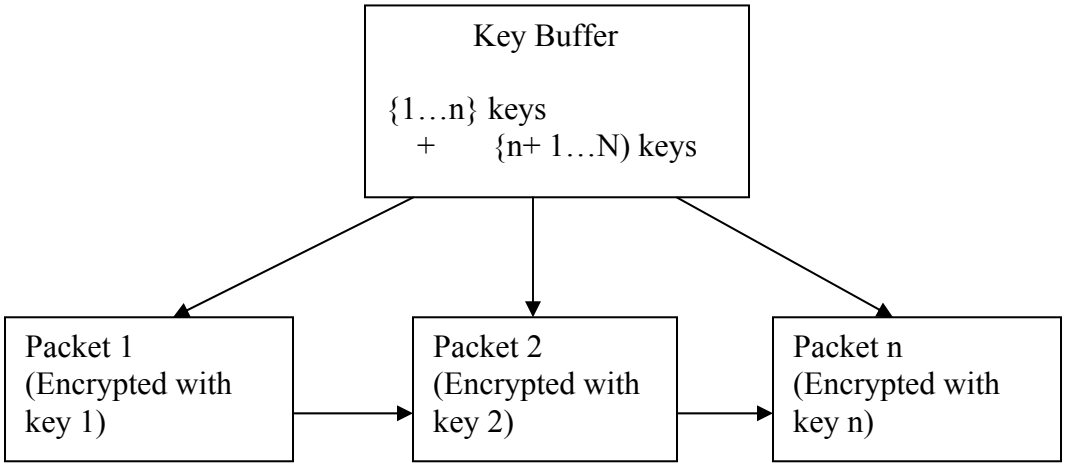


Figure 11 : Packet Encryption in Alternate Encryption Scheme

4.2 Random Number Generation

. Random numbers to be used as keying material in the proposed scheme have to be generated securely as the security of the proposed scheme depends mainly on the cryptographic keys used. The RFC1750 [16] defines various methods and considerations that are required for generating random numbers in both hardware and software environments. Generally for hardware based implementation the random numbers are generated from thermal noise, RF noise high resolution timings of environmental events, or other unpredictable physical sources of entropy [3]. For the software based random number generators the sources include system clocks, system or input/output buffers, user/system/hardware/network serial numbers and/or addresses and timing, and user input [16].

The RFC1750 [16] recommends use of multiple random inputs with a strong mixing function that can overcome weakness in any particular input. Cryptographically secure random numbers can also be generated from random seeds using a generator algorithm. The software

based random number generator implementation differs for each operating system. They are generally based on generating random numbers from system calls and other operating system functions, and are then mixed with a strong mixing function.

4.3 MIKEY Key Exchange mechanism

A secure key exchange mechanism is essential for the proposed encryption scheme to work efficiently. IKE and MIKEY are the two widely used key exchange mechanisms. IKE is used mainly for IPSec traffic and MIKEY was developed for peer to peer security protocols for real-time traffic such as SRTP.

MIKEY Key exchange mechanisms can be broadly classified as pre-shared, PKI based and Diffie-Hellman based key exchange mechanism.

- Pre-shared method - In pre-shared key exchange method the keys are exchanged between the peers before the start of the communication and is based on symmetric cryptography. This type of key exchange does not require much data exchange for establishing the keys. The disadvantage of pre-shared mechanism is that not very scalable when used for many peer to peer communications simultaneously.
- Diffie-Hellman Method - In the Diffie-Hellman key exchange method the communicating parties generate a common shared secret through a series of exchanges. The DH key exchange mechanism is computationally intensive and consumes more bandwidth than the pre-shared mechanism. The DH key exchange mechanism is also susceptible to man-in-the-middle attacks. Therefore it needs PKI or other authentication mechanism.
- Public-Key method - The Public-Key method is based on Public-Key cryptography. It requires PKI architecture, which is resource and time consuming but provides more secure authentication than the previous methods. In the Public-Key based key exchange mechanism

used for the proposed scheme the initial authentication is achieved by the digital signature method based on the RSA algorithm. In the digital signature method the sender sends the security parameters such as the encryption algorithm, its certificate and other information encrypted with the recipient's Public-Key. This information can be decrypted only with the recipient's private key. In the proposed scheme the Public-Keys of the users are stored in a directory by the service provider and can be retrieved only by sending a request to the service provider before initiating a call. This provides protection against MITM attacks.

The PKI architecture requires the users to register for certificates. The certificates are issued and managed by RAs and CAs and are used to authenticate the users. The CAs and RAs are trusted third parties who manage certificate issuance, storage and revocation. In the proposed scheme the service provider can appoint these trusted CAs. Consider a communication scenario between two users Alice and Bob,

1. The initiator Alice sends a REQ message to the recipient Bob for start of the communication session. The Request message is signed with Alice's private key.
2. The service provider/proxy server upon receiving the message checks the validity of Alice's certificates and forwards the message to Bob.
3. Bob decrypts the message using Alice's public key obtained from the service provider's directory. Bob then sends an ACK message back to Alice.
4. The security parameters to be used in the communication session are negotiated in the following messages before the start of the session.
5. After the authentication using the above PKI architecture the symmetric keys to be used in the scheme are exchanged between the users by enclosing it in a digital envelope encrypted with the other user's Public-Key.

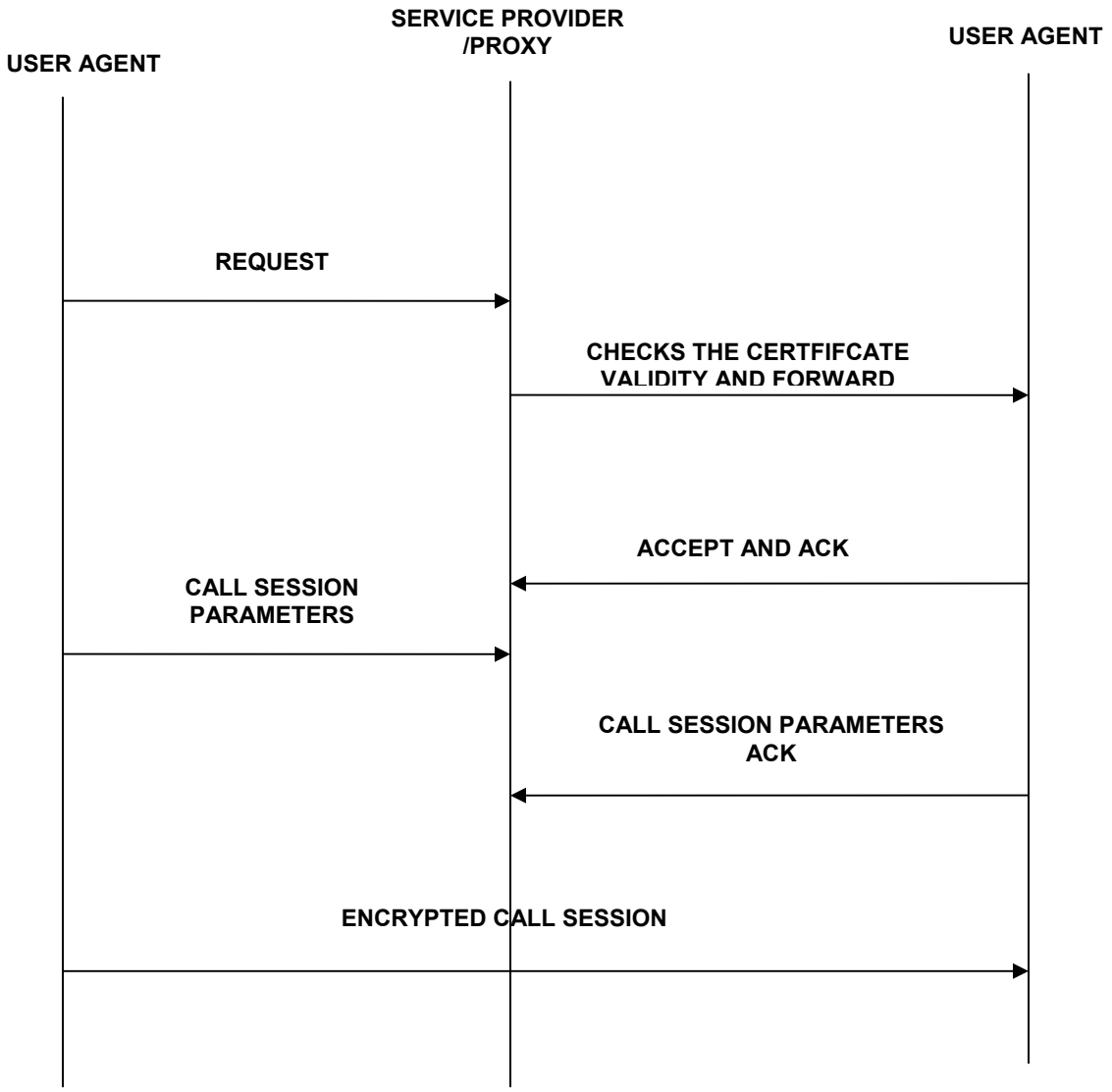


Figure 12: Call Setup Flow diagram for Alternate Encryption Scheme

4.4 Packet Processing in Alternate encryption scheme

After the initial authentication and key exchange using the PKI architecture the encryption process take place as defined below,

1. The index of the RTP packet is determined as in SRTP ,

i = 2^16 * ROC + SEQ.

2. The key for the first packet index of the session is calculated as the first 128 bits of the keying material negotiated between the communicating users and so on.
3. After determining the keying material based on the packet index. The RTP payload is encrypted using the encryption algorithm negotiated between the users before the start of the session.
4. The ROC is updated if needed and sends the packet through the network to the receiver.
5. The receiver on receiving the packet calculates the packet index. This calculation is similar to the receiver side implementation in SRTP.
6. The cryptographic algorithm to be used is determined from the previously negotiated security parameters.
7. The key to be used with the packet is retrieved based on the packet index. The decryption key derivation also depends upon the cryptographic algorithm.
8. The payload is decrypted using the decryption key and the cryptographic algorithm specified in the security parameters.

4.5 Packet Index and ROC

The packet index calculation is similar as in the SRTP implementation. The users maintain a ROC (Rollover counter) which is initialized by the sender to zero each time a session starts. Each time the RTP sequence number, SEQ, wraps modulo 2^16, the sender side must increment ROC by one, modulo 2^32 [2]. The sender side packet index is then calculated using the RTP sequence number and ROC as**, i = 2^16 * ROC + SEQ**. On the receiver side the packet index is calculated as, **i = 2^16 * v + SEQ** where v is the value chosen from the set {ROC-1, ROC, ROC+1} (modulo 2^32) such that i is closest (in modulo 2^48 sense) to the

value 2^16 * ROC + s_l. More detailed information about the SRTP packet index calculation process can be found in the RFC 3711 [2]. The packet index calculation depends on the users requirements like the level of tolerance to packet loss and level of synchronization.

4.6 Security Parameters

Before the start of a communication session the network and security parameters has to be negotiated. For a RTP session, the destination addresses and ports of the communicating users are first negotiated. The proposed alternate encryption scheme also negotiates the cryptographic algorithm to be used, the key sizes to be used and in some cases it can also include the duration of the media sessions in order to determine the number of keys required for a particular session. A packet with no security parameters information negotiated is discarded.

4.7 Cryptographic algorithms

The proposed alternate encryption scheme uses Public-Key cryptographic algorithms for the initial authentication and key exchange mechanism. The MIKEY key exchange mechanism based on RSA can be used. For the actual payload encryption the alternate encryption scheme uses symmetric cryptographic algorithms. The scheme proposes to use an encryption algorithm that requires less computational power and produces less delay. Such an ideal cryptographic algorithm is yet to be developed. Based upon the comparison of the widely used symmetric algorithms AES can be used for the proposed scheme.

4.8 Authentication Mechanism

The proposed alternate encryption scheme accomplishes initial authentication of the users using the PKI architecture. The use of an authentication mechanism for each packet has not

been analyzed by this work. But the communicating users may use an authentication mechanism such as the HMAC-SHA1 and calculate the authentication tag for each packet.

CHAPTER 5

PERFORMANCE ANALYSIS OF THE ALTERNATE ENCRYPTION SCHEME

*5.*1 Delays

The performance of any security protocol VOIP or other real-time applications is measured on the basis of the delays. The total acceptable delay for a VOIP packet is only 150ms. The delays for a VOIP session can be classified as the Call setup delay that happens before the actual call and the delays during call. The call setup delay generally consists of the signaling delays caused by signaling protocols such as SIP or H.323 and the initial key exchange delay. The delays possible during call are the encryption delay, decryption delay and network delays. The delays in VOIP can be summarized as,

Call Delay = T cs + T ke + T enc + T dec + Network delays

Where, T cs is the call setup delay, T ke – key exchange delay, T enc – encryption delay and T dec – decryption delay. This research work mainly concentrates on the delays during the call. The following sections compare the per-packet encryption and decryption delays caused when using SRTP and the proposed Alternate encryption scheme.

5.2 Call Setup Delay

The call setup delay of a VOIP packet involves the delay caused for setting up the signaling based on protocols such as SIP or H.323 and the initial key exchange and authentication phase. The SRTP protocol doesn't define a key exchange or signaling mechanism of its own. The ZRTP is an extension to SRTP and defines a key exchange mechanism based on Diffie-Hellman mechanism.

The initial authentication and key exchange mechanism in the proposed Alternate Encryption Scheme might be longer than the call setup time of SRTP or ZRTP due to the use of PKI in the proposed scheme. The time taken for the initial authentication scheme has not been considered in this research work.

5.3 Delays during Call

The performance of the proposed scheme has been analyzed by estimating the delay caused during the call i.e., the delay for the media encryption and decryption. The call setup delay has not been considered for this research work. The major delays caused during the media protection in SRTP are the encryption delay, decryption delay and the key generation delay. The delays can be expressed as,

Per-Packet Delay libSRTP = T enc + T dec + T kg + T rp

And for the proposed alternate encryption scheme, the delay during call is expressed as,

Per-Packet Delay Alt. Encryption Scheme = T enc + T dec + T kf

Where, T enc represents the delay caused due to encryption, T dec – delay due to decryption , T kg – delay for key generation for particular packet , T rp – delay for replay protection and T kf – delay for fetching the key. The key fetch delay is assumed to be negligible in the case of the proposed protocol and is neglected.

5.3.1 Per- Packet Encryption and Decryption delays

The encryption and decryption delays of the proposed Alternate Encryption Scheme has been simulated and calculated in this research work using libSRTP. The libSRTP is an open-source implementation of SRTP [17]. The initial implementation of libSRTP was done by David McGrew of Cisco systems [17]. The source code for the libSRTP has been implemented in C.

This libSRTP supports all of the mandatory-to-implement features of SRTP [17]. Some options that are described in the SRTP specification are not supported. This includes the Master Key Index (MKI), key derivation rates other than zero, the cipher F8, anti-replay lists with sizes other than 128 and the use of the packet index to select between master keys [17]. The simulations have been run on Linux operating system. The time delay values for the SRTP_Protect [17] , SRTP_Unprotect [17], Replay protection and the time taken for generating encryption keys and authentication keys has been calculated. The SRTP_Protect [17] function of the libSRTP implementation encrypts the RTP packets based on the policy defined turning it into SRTP packets. Similarly the SRTP_Unprotect [17] function decrypts the SRTP packet and returns the original RTP packet.

Timestamp code has been inserted into the libSRTP implementation to calculate the approximate time taken by the SRTP_Protect [17] and SRTP_Unprotect [17] function for each packet. These values give the approximate amount of time taken by SRTP protocol to encrypt and decrypt a packet. The time delay for providing replay protection and generating the encryption and salt keys have also been calculate by inserting timestamp hooks at appropriate portions of the libSRTP source code. The following data has been tabulated by running the srtp_driver [17] test program and calculating the approximate time taken for encrypting and decrypting the test packets. The various combinations tried out are Null Cipher and Null Auth, AES ICM and Null Auth, Null Cipher and HMAC-SHA1, AES ICM and HMAC-SHA1.

The simulations have been run many times, and the average mean value of time-delay has been recorded. The simulations were run on a Linux machine and the receiver and sender were simulated using two terminals one for sender and one for receiver.

	SRTP (ms)	Alt. Encryption Scheme (ms)
Null Cipher and Null Auth	1.107	0.952
Null Cipher and Hmac-Sha1	3.320	2.581
AES ICM and Null Auth	3.915	2.762
AES ICM and Hmac-SHA1	7.355	5.582

Table 2: Per- Packet Encryption Time Delay

The values in the table are approximated to milliseconds. The encryption time delay for the proposed Alternate Encryption Scheme has been calculated as the difference of the encryption delay of SRTP and the value of generating encryption and authentication keys for SRTP. Since the proposed protocol doesn't require any key derivation for encrypting each packet, the difference value obtained approximately represents the time taken by the proposed scheme.

The time delay for the libSRTP to provide replay protection and generating the encryption and authentication keys for each SRTP has been approximately calculated by inserting timestamp hooks at appropriate portions of the libSRTP source code. Note that the encryption delay estimated as above is only based on per-packet and does not consider the key fetch time for the packet in the proposed protocol. Also the time delay estimates are based on the processor speed of the system used. The decryption delay for each packet is also calculated using the same process defined above at the receiver side. The time taken for the SRTP packet to initialize , to check for replay protection and the time taken for SRTP_Unprotect [17] function

approximately represents the decryption time delay for each packet in libSRTP. The table below represents the time delay for decrypting a packet using libSRTP and the proposed protocol.

	SRTP (ms)	Alt. Encryption Scheme (ms)
Null Cipher and Null Auth	1.32	1.087
Null Cipher and Hmac-Sha1	4.027	2.872
AES ICM and Null Auth	4.21	3.056
AES ICM and Hmac-SHA1	7.984	5.582

Table 3: Per-Packet Decryption Time Delay

The graphical representation shows the comparison of the per-packet encryption and decryption delays of SRTP and the Alternate Encryption Scheme. The graphs show a notable decrease in the time taken for the per-packet encryption and decryption using the proposed protocol.

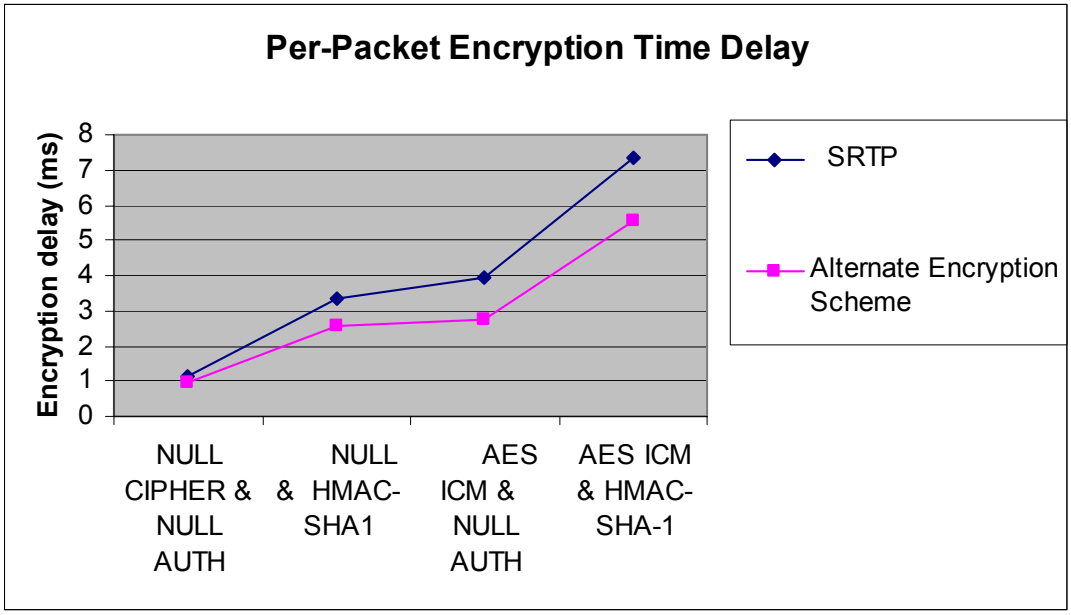


Figure 13: Per- Packet Encryption Delay

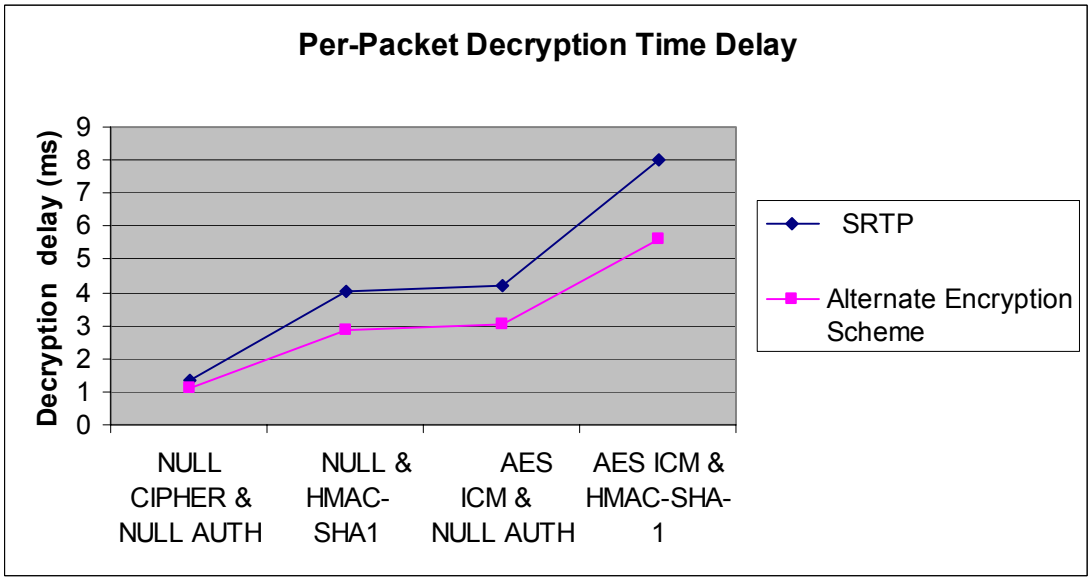


Figure 14: Per-Packet Decryption Delay

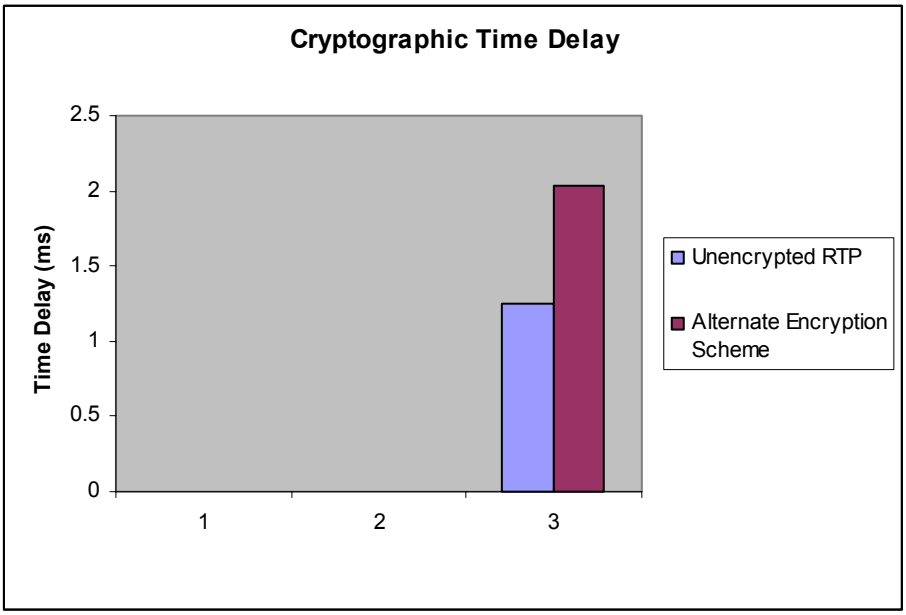


Figure 15 : Comparison between Unencrypted RTP and Alternate Encryption Scheme

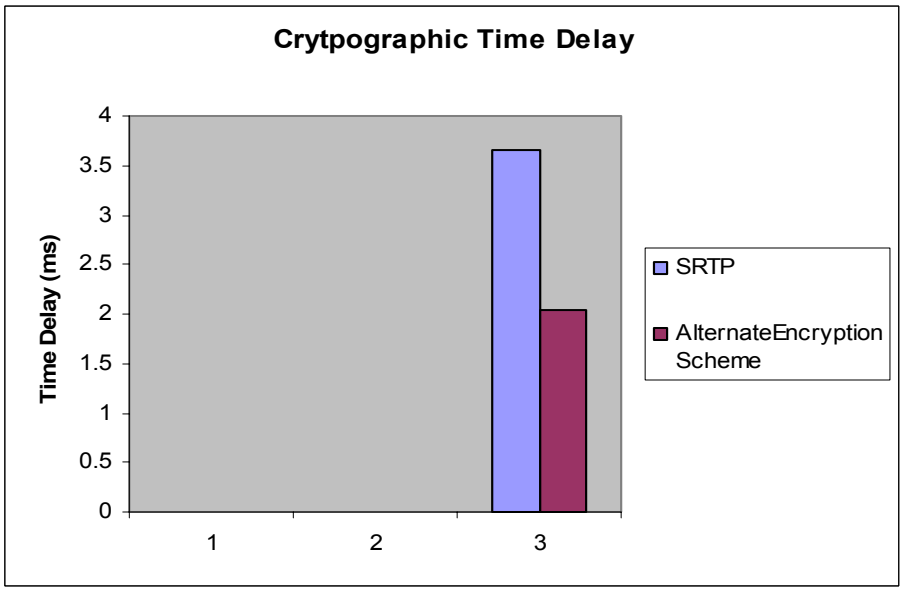


Figure 16 : Comparison between SRTP and Alternate Encryption Scheme

The figures 14 and 15 show the cryptographic time delay caused due to Alternate Encryption Scheme in comparison with Unencrypted RTP and SRTP.

5.4 Brute Force Attack

The most common type of attack on a secured session is the Brute force attack. The strength of the proposed Alternate encryption scheme against the brute force attack is examined in this section. Consider 'n' keys shared between the caller and callee. A hacker trying to wiretap the communication will have to open a brute force attack. In the proposed Alternate Encryption scheme the hacker has to attack or decrypt each packet with a unique key. Hence the decryption of the whole voice stream becomes very difficult. The brute force attack can be mathematically illustrated as follows, Let T be the time needed to decrypt the voice session which is encrypted by the normal encryption technique (using a single session key for the whole session).

$$T = (2^{L}) * C \quad \text{--------- (1)}$$

Where,

L represents the length of the key.

C represents the time needed to run the algorithm once with a key

The time needed to decrypt the voice session encrypted using Alternate encryption scheme would be,

$$T1 = (2^{LN}) * C1 \quad \text{---------- (2)}$$

Where,

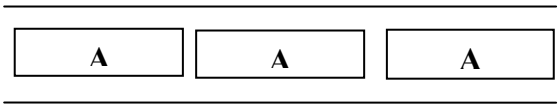
N represents the number of keys

L represents the length of the key

C1 represents the time needed to run the algorithm once with a key

In the encryption schemes using session keys the hacker can use the brute force attack by trying to predict single session key used for encryption. In the proposed alternate encryption scheme the hacker has to use a unique decryption key for each packet in order to decrypt the voice session.

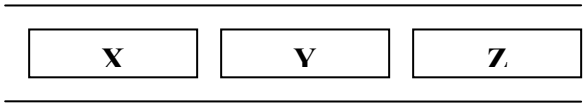
Session with a single key



A – Key of length (l)

Number of combinations = 2^{L*1} (Since only one key of same type is used for the whole session)

Session with three keys



X, Y, Z – Different keys each of length (l)

Number of combinations = 2^{L*3} (Since three different keys are used). When, N number of keys are used the number of combinations is 2^{l*N}.

5.5 MiTM Attacks

One of most common attacks on a VOIP session is the man-in-the-middle (MiTM) attacks. Man-in-the-middle attack (MITM) is an attack in which an attacker is able to read, insert and modify at will, messages between two parties without either party knowing that the link between them has been compromised [18]. Consider a scenario where two communicating parties Alice and Bob want to establish a communication session between them, for establishing the session Alice requests Bob for his Pubic key. If a hacker Eve is able to intercept the request and pass his Public-Key to Alice pretending to be Bob then MiTM attack takes place. In this

attack the hacker Eve acts as Bob to Alice and as Alice to Bob and can capture the entire voice session.

In the Alternate Encryption scheme protection against MiTM attacks if provided by the PKI architecture. In the proposed scheme before establishing a session Alice sends a request to the service provider, the service provide then verifies the Certificate of Alice and if found genuine sends the Public-Key of Bob to Alice. The Public-Keys are stored in directories that can be accessed only by the service provider. Hence in the proposed Alternate encryption scheme MiTM attacks are not possible.

5.6 Advantages of the Proposed Scheme

Over SRTP

The proposed alternate encryption scheme uses a unique key for each packet and encryption is done on a per packet basis. The SRTP protocol described before uses session keys to encrypt packets of a particular session. The session keys are derived from a single master key. A leaked master reveals all the session keys derived from it – single point of failure. Also a cryptanalyst collecting large data encrypted using a single session key can plant some kinds of attacks. In the proposed algorithm the encryption keys are unique for each packet, even when a cryptanalyst captures a large part of cipher text he cannot get any information from it since the keys used have relation to each other.

Over ZRTP

ZRTP protocol is similar to SRTP in the aspect of media encryption part. So the weaknesses of SRTP described above hold good for ZRTP also. ZRTP is basically an extension to SRTP and proposes a key exchange mechanism to be used with SRTP based upon Diffie-Hellman scheme.

The major issue in the suggested Diffie-Hellman scheme is the MITM attacks. The ZRTP doesn't use any PKI architecture and relies upon SAS mechanism to counter the MITM attacks. Inattentive users who skip the SAS mechanism may pave way for the MITM attacks. Also there are some kinds of voice spoofing attacks that ZRTP is vulnerable to. The proposed alternate encryption scheme uses Public-Key cryptography for the initial key exchange process. Also the proposed algorithm suggests use of full fledged PKI architecture to provide protection against MITM and other attacks.

5.7 Disadvantages of the Alternate Encryption Scheme

The major concern with the proposed algorithm is setting up and maintaining the PKI. Maintaining the PKI involves a lot a work such as issuing, verifying and revoking the certificates. The secure storage of the Public-Keys and the certificates is also important. In the proposed scheme the above concerns are meant to be addressed by the service provider. Another important issue in the proposed scheme is the generation of random numbers. Since each packet is encrypted with a unique key attention has to be given to generation of random numbers. The section 4.2 explained the various the methods available for random number generation. The users or the service provider have to select a suitable method based upon the needs.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The research work provided a method for securing the Real-time traffic in general and VOIP in particular. The Alternate Encryption Scheme provides better security than the existing security protocols such as SRTP and ZRTP. But the call-setup and initial authentication delays are suspected to be higher. The Alternate encryption scheme suggests reducing the computational time of the underlying encryption algorithm by using an algorithm that is of lesser complexity and takes less computational time. Such an algorithm has to be developed in the future. This research work has simulated and analyzed only the time delays during the call, the initial call setup delays and the network delays have not been considered. Further research needs to be done to simulate the delays caused during call setup and the ways to minimize them.

The initial authentication of the proposed Alternate Encryption scheme relies on PKI architecture. Developing and maintaining PKI architecture is an intensive task. Future work needs to be done to develop a suitable PKI architecture that is efficient for the proposed scheme. Efficient ways to generate and manage keys for the proposed scheme has to be analyzed. Efficient ways to maintain the key buffers and the key fetch delays have to be analyzed. In this research work the proposed protocol has been simulated using only AES encryption algorithm. The delays caused when using other encryption algorithms can be evaluated.

The operation of the Alternate Encryption Scheme protocol can also be analyzed by using Non-PKI architecture for the initial key exchange. Since maintaining the PKI architecture is intensive alternate ways for the initial authentication can be explored. The proposed Alternate Encryption scheme is not CALEA compliant.

Work has to be done in order to make the proposed scheme CALEA complaint. Also the additional overhead and delay that would be added by making it CALEA compliant has to be evaluated.

REFERENCES

LIST OF REFERENCES

[1] H Schulzrinne, S Casner, R Frederick and V Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003, http://www.ietf.org/rfc/rfc3550.txt

[2] M Baugher, D McGrew, M Naslund, E Carrara, and K Norrman, "Secure Real-time Transport Protocol," RFC 3711, March 2004, http://www.packetizer.com/rfc/rfc3711

[3] P Zimmermann, A Johnston, and J Callas "ZRTP: Media Path Key Agreement for Secure RTP," Internet Draft, March 2007,http://www.ietf.org/internet-drafts/draftzimmermann-avt-zrtp-03.txt

[4] Zfone. The Zfone Project. http://zfoneproject.com/, 2006

[5] J Orrblad, "Alternatives to MIKEY/SRTP to secure VOIP," Masters Thesis, Telecommunication Systems Laboratory, Dept. of Microelectronics and Information Technology, Royal Institute of Technology, KTH, Stockholm, Sweden, March 2005

[6] J Bilien, E Eliasson, J Orrblad, J-O Vatn, "Secure VoIP: call establishment and media protection," In the Proceedings of 2nd Workshop on Securing Voice over IP, Washington DC, June 2005

[7] D R Kuhn, T J Walsh, and S Fries, "Security Considerations for Voice over IP Systems," Special publication, 800-58, National Institute of Standards and Technology, January 2005

[8] C Jie, "Design Alternatives and Implementation of PKI Functionality for VOIP," Masters Thesis, Telecommunication Systems Laboratory, Dept. of Microelectronics and Information Technology, Royal Institute of Technology, KTH, Stockholm, Sweden, June 2006

[9] "RSA Algorithm," Technical Document, http://www.rsa.com/rsalabs/node.asp?id=2146

[10] Menezes Alfred, Van Oorschot Paul C and Vanstone, Scott A. "Handbook of Applied Cryptography". CRC Press, October 1996

[11] Matt J. B. Robshaw, "Stream Ciphers Technical Report TR-701, version 2.0, RSA Laboratories", July 1995

[12] M. Liskov, R. Rivest, and D. Wagner, "Tweakable Block Ciphers", Crypto 2002

[13] http://www.tropsoft.com/strongenc/des3.htm

[14] J Kelsey, S Lucks, B Schneier, M Stay, D Wagner, and D Whiting, "Improved Cryptanalysis of Rijndael", Fast Software Encryption, 2000

[15] R. Housley, “Using Advanced Encryption Standard (AES) Counter Mode with IPSec Encapsulating Security Payload (ESP)”, RFC 3686, January 2004, http://www.ietf.org/rfc/rfc3686.txt

[16] D Estalake, S Crocker, and J Schiller, “Randomness Recommendations for Security,” RFC 1750, December 1994,http://www.ietf.org/rfc/rfc1750.txt

[17] LibSRTP documentation, http://srtp.sourceforge.net/srtp.html

[18] D N Sarpanos, R J Lipton, “Defense Against Man-in-the-Middle Attack in Client-Server Systems,” In the Proceedings of Sixth IEEE Symposium on Computers and Communications, Hammamet, Tunisia, July 2001

[19] http://www.di-mgt.com.au/rsa_alg.html

[20] Jothy Rosenberg, David Remy, “Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption”, Sams, 2004.