European Journal of Scientific Research
ISSN 1450-216X Vol.40 No.2 (2010), pp.177-188
© EuroJournals Publishing, Inc. 2010
http://www.eurojournals.com/ejsr.htm

SVS - A Secure Scheme for Video Streaming Using SRTP AES and DH

Mamoona Asghar
Department of Computer Science and IT, The Islamia University of Bahawalpur
Punjab, Pakistan
E-mail: tweety_mees@yahoo.com
Tel: +92-62-9255466

Saima Sadaf
Department of Computer Science and IT, The Islamia University of Bahawalpur
Punjab, Pakistan
Tel: +92-62-9255466

Kamran Eidi
Department of Computer Science and IT, The Islamia University of Bahawalpur
Punjab, Pakistan
Tel: +92-62-9255466

Asia Naseem
Department of Computer Science and IT, The Islamia University of Bahawalpur
Punjab, Pakistan
Tel: +92-62-9255466

Shahid Naweed
Department of Computer Science and IT, The Islamia University of Bahawalpur
Punjab, Pakistan
Tel: +92-62-9255466

Abstract

Video streaming technologies are becoming immensely important with the growth of the multimedia technology. With streaming, the end user can start watching the file almost as soon as it begins downloading. Security becomes a key problem to be handled when your valuable multimedia assets are floating over the network. This paper presents the key management and encryption mechanisms on the Video (Motion Picture + Sound) streams on un-trusted client-server network. As Network data Security is a burning issue and dominates the communication systems today, we present a novel security scheme, named as SVS (Secure Video Streaming) to securely transfer the valuable multimedia video streams on the un-trusted network using the authorization, key management, encryption, packetization and authentication schemes. The process of authorization is done by allowing the network access to only authorized users, keys are generated through DH (Diffie-Hellman) key exchange mechanism, AES (Advance Encryption Standard) algorithm is used for data encryption, and then encrypted data is embedded into the SRTP (Secure Real-Time Transport Protocol) header. The SRTCP (Secure Real-Time Transport

Control Protocol) Sender and Receiver reports are also generated for data acknowledgement. A keyed-hash algorithm is used to generate the MAC (Message Authentication Code) for every SRTP packet. After undergoing all these processes, the data is finally traveling over the network media. As a result, we hopefully tend to apply the best possible security on multimedia streams traveling among the Networks.

Keywords: AES, Authentication, Authorization, Cryptography, DH, MAC, Packetization, SHA-1, SRTP, SRTCP, UDP, Video Streaming.

1. Introduction

Streaming visual data to different users is becoming increasingly popular in recent times, and protecting the transmitted data from every possible security threat has become one of the main concerns both for the end users and data providers. This paper describes a method for protecting streamed data from possible security attacks and suggests a design of secured system architecture for multimedia video streaming to multiple receivers (one receiver at a time) considering the state of the art for the video streaming existing today. The main feature of the suggested design is its ability to provide a secure communication environment for real-time data.

For key generation, the Diffie-Hellman algorithm[1] is used. This key exchange algorithm is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communication channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

A computer security standard Advance Encryption Standard (AES) [2] is used for the encryption and decryption of data where the cryptography scheme is a symmetric block cipher that encrypts and decrypts data using 128 bit key. As an efficient encryption standard, it is currently being deployed on a large scale.

SRTP (Secured Real-Time Transport Protocol) [3] is used for packetization of data as well as to enhance its security. In this scheme we first construct the SRTP packets by embedding the video data and then transmit them on the network media. SRTP (a profile of RTP) which can provide confidentiality, message authentication, and replay protection to the RTP traffic and to the control traffic of RTP(Real-Time Transport protocol) and RTCP (Real-Time Transport Control protocol).[4] A keyed-hash function is used to compute the MAC (Message Authentication Code) for every SRTP packet to authenticate the data traveling across the network.

The technique of multithreading is implemented in this research work to handle multiple receivers simultaneously, thus the users do not need to wait for the server for getting registered or login, or for sending their request to the server. Server is responding fast to each of its user using the multithreading techniques such that each user feels that he/she is interacting with the server alone.

We organize the remaining of the paper as follows: section II describes the streaming principle and related work; section III and IV presents our network and application architectures respectively; section V is the Implementation section which presents briefly how the security problems have been handled in our scheme. The security threats and their handled solutions are also discussed there, section VI presents our results in detail; section VII states our conclusion and flexibility of this article for adapting to any future work.

2. Video Streaming

In streaming mechanism, the file is sent to the end user in a (more or less) constant stream. It is simply a technique for transferring data such that it can be processed as a steady and continuous stream and it is called Streaming. Streaming video is a sequence of "moving images" that are sent in compressed form over the Internet and displayed by the viewer as they arrive. If a web user is receiving the video data as streams then he/she does not have to wait to download a large file before watching the video or listening to the audio**.**

Streaming Principle: Streaming multimedia is a multimedia streams that is constantly received by, and normally rendered to, the end-user screen while it is being delivered by the provider. In streaming applications it is necessary for the data packets to reach their destination in a timely manner because the delay can cause the network congestion, and can result in the loss of all those packets suffering from excessive delay. This causes loss of quality of data, the synchronization between client and server to be broken, and errors to propagate in the rendered video.

There are two types of steaming, one is real time and other is prestored or prerecorded streaming. The scheme we have developed can work for both real time and prestored video files. The protocol used for streaming purpose is UDP (User Datagram Protocol) [5] which sends the media stream as a series of small packets. This is simple and efficient; however, there is no mechanism within the protocol to guarantees delivery because it does not acknowledge to the sender, but in SVS application the SRTCP reports have been specifically use for the acknowledgement to the sender continuously about the delivery of data at the destination.

3. SVS Network Architecture

Basic network architecture of SVS application is client-server based. As today most of the communication is Unicast and Broadcast transmission based because group communication is not needed everywhere. SVS scheme is practically a unicast network architecture, but apparently seems to be the broadcast network because threads handle all the clients simultaneously. This article presents the network with single server and multiple clients where all clients are interacting with the server alone for making requests, and due to the implementation of multithreading the interaction among server and clients is so fast and in such a managed time slices that each client feels that he/she is the only person whom server is responding to.

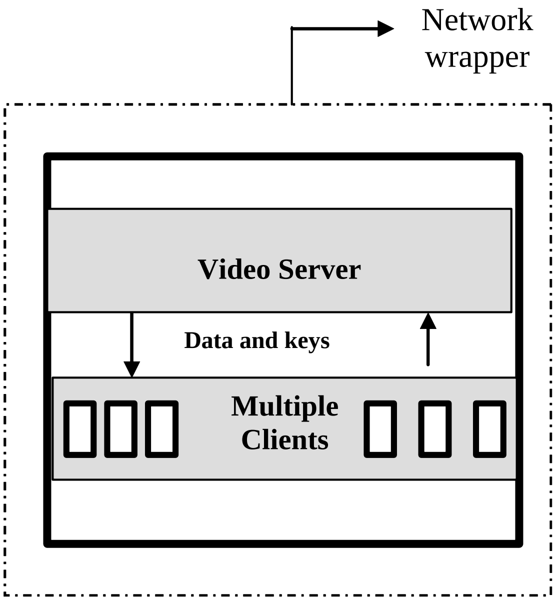
The Server and the client of SVS architecture remain in contact continuously during one session. There are two different communication scenarios for the users who are already member of the server or who are still unregistered. For registered users the server first verifies the clients and if they are really authorized, it sends them the same unique encrypted key for their particular session called Session Key (SK), after that client requests for any available video file to server and multimedia streams start transmitting from the server to the client.

For unregistered users the mechanism is a bit different, they first need to get registered with server by sending some of their confidential information. The server allocates them a unique life time key called Master Key (MK), and after getting that key the client becomes able to communicate with the server as an authorized user through the same way as described earlier. It is to be noted here that the mechanism of exchanging the keys during registration or verification process is totally secure which will be explained next in the concerning portion of this article.

There are two major implemented modules of this network architecture:

1. Video Server
2. Client

Figure 1: SVS Network architecture



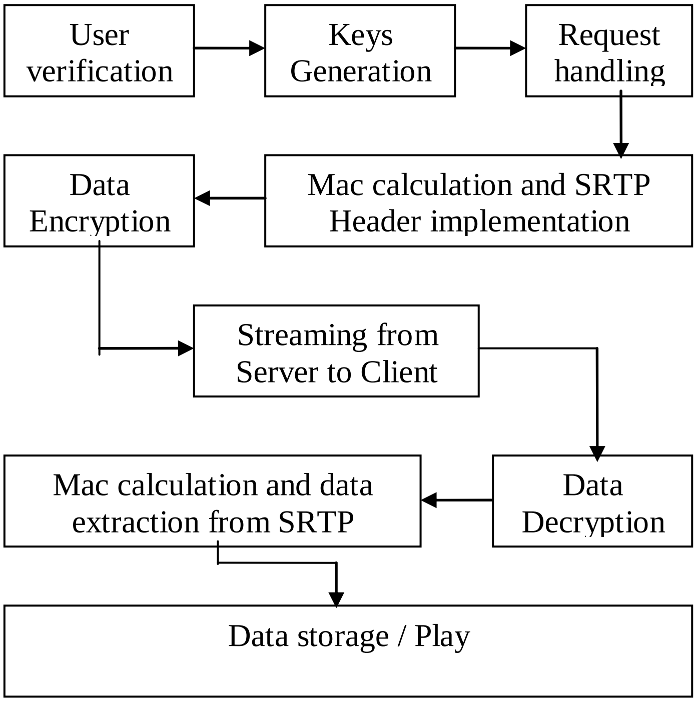
The responsibilities of Video Server are to register or authorize users, to verify their identity, to allow them to make request for data they require, to handle requests made by clients and finally to send them the required data but only after necessary processing. Another aspect of this application is that the server detects the IP addresses of all clients who interact with it and then maintains a log of all those clients who join or leave it.

Once the client get registered to Video Server , it can avail all the features of application, it can ask for any available video file from server and can get it with full confidentiality and without any fear of data theft on the network.

4. SVS Application Architecture

The application architecture or communication flow among server and client of SVS application is simple and secure. Security is implemented using the combination of such mechanisms which are capable of providing protection against all possible attacks on data, and at the same time the implementation is in such a way that it creates minimum of traffic across the network by reducing the necessity of two way transport of video streams. SVS application is not producing un-necessary keys; only two keys are generated to achieve the necessary security.

Figure 2: SVS Application Architecture


User verification
Keys Generation
Request handling
Mac calculation and SRTP Header implementation
Data Encryption
Streaming from Server to Client
Data Decryption
Mac calculation and data extraction from SRTP
Data storage / Play

5. Implementation

The multimedia is likely to play an important role in the future. When data travels across the network, its security becomes the most important issue because in this era of advance technologies data theft has become increasingly common and data privacy is unsafe, that’s why data security demands more attention. Data and network security is recognized as critical issues to the design of modern communication systems.

The Network security consists of the provisions made in an underlying computer network infrastructure and it starts from authenticating any user. Once authenticated, security techniques enforce access policies and decide what services are allowed to the network users.

UDP (User Datagram Protocol) Streaming is used in SVS application; UDP is often the favored transport protocol for delivery of audio and video data over IP networks because of flow overhead. However, as UDP provides no guarantee of delivery, a video streaming application needs to be provided with extra information to enable the detection of lost, delayed and unordered packet delivery. UDP is usually supplemented with framing and feedback information, such as that provided by the Real-Time Transport Protocol (RTP) and the associated RTP control protocol (RTCP). [6] So even though UDP is often the transport protocol of choice for many streaming applications it provides no acknowledgement because this is intended for use where timely delivery is higher priority than overall reliability. [7] UDP’s approach is ‘send it and forget it’, so in SVS application, in order to ensure the reliability too, the UDP is used with SRTCP streams to get the Receiver reports. So the use of SRTP with its additional sequencing and loss detection information in conjunction with UDP provides a mechanism for the application to ensure that packets are used in the correct order and occurrences of loss of packets are detected.

There are a many standardization bodies, which are concerned with developing and improving protocols for the transport of real-time data over IP networks — the international Telecommunications Union (ITU), International Standards Organization (ISO) and the Internet Engineering Task Force (IETF), among many others. Such protocols have the goal of compensating for the lack of features in the classic IP protocols, such as UDP.[7] We use SRTP and SRTCP for the loss detection mechanism that is missing in UDP as well as providing content identification, sequencing and timing information.

The SVS application has four major phases which together are ensuring the secure data communication among the network:

5.1. User Authorization

Authorization allows you to set parameters that restrict a user to access network services. It is a key phase to make network security better. In SVS application user authorization has given main attention. To access the network one needs to have the membership of server and after that, the user is required to login each time he/she needs some data from the server.

For being the member of the server, one needs to provide some of its confidential information to server like Name and Identity card number etc, Server stores all the information, maintains a complete log of all registered users and each time they try to login, they are verified by their existing information like user name, password. Each registered user has unique identification records and is provided with a unique Master Key (derived through Diffie Hellman) at the time of registration and a new unique but random Session Key at the time of login for a particular session.

The algorithm we are using for the key generation is “Diffie-Hellman key exchange algorithm”. This, by itself does not provide authentication of the communicating parties and is thus vulnerable to a man-in-the-middle attack. So for avoiding this type of attack the MAC (Message Authentication Code) is calculated in every SRTP packet, which provides a best way for compensating this inherent deficiency of Diffie-Hellman exchange.

5.2.Key Management

Keys management is one of the major requirements for data security. Appropriate and successful key management is critical to the secure use of every crypto system without exception. Key transport methods may be different in different scenarios. Two major methods for transporting or establishing the major keys are pre-shared key and public key method, the Diffie-Hellman key exchange method which is used by SVS is a public key method.

In practice, most attacks on public-key systems will probably be aimed at the key management level, rather than at the cryptographic algorithm. In SVS application Key management deals with the secure generation, distribution, and storage of keys. We believe that Diffie-Hellman (DH) key agreement method is the most secure method because in it the final keys both of sender and receiver are never travel on the network so they are unavailable to the third party eavesdroppers [1].

Users must be able to store their private keys securely, so that no intruder can obtain them, yet the keys must be readily accessible for legitimate use. Keys need to be valid only until a specified expiration date but the expiration date must be chosen properly.

Both of these requirements have been achieved in SVS scheme. The actual Master key never travels across the network; there is no possibility of its interception, so its security is confirmed. Further that the Master key is used to encrypt the Session key of individual by using the Advance Encryption Standard (AES). Session keys travel through network from server to client but only after their encryption. Those session keys are then used for the encryption/decryption of multimedia. Validity of the session key is only for one session, so for each new session, the session key will be different.

Figure 3: SVS Keys Summary

SVS Overall Key Management:

The following **keys are derived**:

1. Master Key (Symetric key of 128 bits)

- Derived and exchanged by Diffie Helman
- Generated & distributed at user registration time.

2.Session Key (Symetric key of 128 bits)

- Derived by using Random number on server side and transfer to client after encryption by Master Key and AES.
- Encrypt the video data by using AES and send it to client.

5.3. Possible attacks on keys and their solution

As there are two major keys in this application because SRTP requires two keys only, Master key and Session Key; our main key is the Master Key which we are driving through Diffie Hellman and as DH allows the two parties to share a secret key without making it travel upon the network, so that key is totally unavailable to any eavesdropper. Second is the Session Key, which is encrypted by the Master key and AES before making it travel across the network, and the first thing we considered for this key is it's validity period. For each new session, the session key will be different even for the same user. So if any intruder succeeds to hack that key, it can not be used for hacking the data because it will be unavailable in the next session (while the current session is already being used by the actual client) and suppose if an un-authorized interceptor captures the key and tries to use it for the current session too then the key which is to be needed for the decryption of the session key (i.e. Master key) will be unavailable for sure because it has never traveled upon the network. So in all cases both keys are unavailable to any un-authorized user. The encryption of sessions keys also avoid the man-in-middle and replay protection attacks for all sessions.

These were some possible attacks and their solution for the Keys only. The attacks on the data are discussed next. In SVS application the crowd of un-necessary keys is totally eliminated and only two keys are used and the way they are used is capable enough to provide the maximum security needed to protect from possible attacks of data theft.

5.4. Encryption/Decryption

Advanced Encryption Standard (AES), also known as Rijndael, is used for the encryption and decryption purpose. AES is a symmetric block cipher protocol (operates on a group of bits (a "block") of a certain length all in one go). [2] The standard key lengths are of 128, 192, and 256 bits. It is efficient and has endured extensive cryptanalytic attacks. AES is therefore a desirable choice and is currently being deployed on a large scale.

And the mode of AES which we are using in SVS application is Advanced Encryption Standard Counter Mode (AES-CM) encryption method, as this encryption method (NULL encryption method also) is mandatory to implement the SRTP. AES-CM is the default encryption method used in SRTP. There is another reason of choosing AES-CM, actually there is no payload expansion produced and the encrypted data is of the same length as the original payload or data. Another feature of AES-CM allows the processing of out-of-order packets, which also implies being able to process packets in parallel. [8]

For the AES algorithm, the length of the input block and the output block is same. It is a point to be noted here that no weak or semi-weak keys have been identified for the AES algorithm and there is no restriction on key selection, only the Key Expansion routine for 256-bit Cipher Keys is slightly different than for 128- and 192-bit Cipher Keys. Here in this application we are using 128 bit key AES, in which there are 10 iterations –called the round key- for being used in the last stage of AES. First three stages are “Sub Bytes”, “Shift Rows” and “Shift Columns. The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the SECRET level. TOP SECRET information will require use of either 192 or 256 key lengths. [9]

In SVS scheme, we provide plaintext (refers to the video data to be encrypted) to AES, it converts it into cipher text, then that cipher text travels upon the network. At receiver side it is deciphered again and brought into its actual form and is stored in the file to be played by user. Design of AES is highly conservative that enables us to demonstrate its security against all known types of active and passive attacks.

5.5. Packetization

Packetization is a simple process of placing video frames into SRTP packets. It is an additional feature of security, the term “SRTP” refers to embed data chunks into a packet following other different data bytes too. Our application is placing the video data into SRTP packets and then SRTP packets are traveling upon the network. SRTP is a profile of RTP. This profile is an extension to the RTP Audio/Video profile. [10] Video data is first encrypted and then placed into the packet, The encrypted portion of an SRTP packet consists of the encryption of the RTP payload (including RTP padding when present) of the equivalent RTP packet. The encrypted portion MAY be the exact size of the plaintext or MAY be larger. [4]

SRTP uses two types of keys: session key and master keys. [3]. Here in SVS application too only two keys are being used, one is called Master key derived from the Diffie-Hellman key exchange method and other is the session key used for the particular session of particular client.

In SVS scheme SRTCP provides a mechanism for providing feedback to the source. It allows monitoring of the SRTP data delivery. Since RTP is closely related to RTCP which can be used to control the RTP session, same as SRTP also has a sister protocol, called Secure RTCP (or SRTCP). It has five types of messages: sender report, receiver report, source description message, bye message, application-specific message. Here we are using three types of SRTCP messages, the SRTCP Sender Report, Receiver Report and BYE message (which will be discussed next). The receiver reports are for passive participants, those that do not send SRTP packets. These reports inform the sender and other receivers about the quality of service. As in SVS application we used UDP streaming which doesn’t guarantee the delivery of packets. In order to overcome this deficiency of UDP, the SRTCP Receiver Reports are used.

Operations of the protocol are carried out under an administrative framework which defines both authentication and authorization policies. SRTP provides a framework for encryption and message authentication of RTP and RTCP streams also the SRTP itself does not provide any mechanism to ensure timely delivery or provide other QoS guarantees, but relies on lower-layer services to do so.

5.6. Message Authentication

Authentication ensures that attackers can neither modify packets in the stream nor insert (forge) additional packets. In addition to the features of key management and encryption techniques, here a key hashing mechanism is used for message authentication called HMAC, which generates the Message authentication Code (MAC) for the data of SRTP header. The authentication operation is performed after the encryption operation and protects the entire RTP packet. Providing a way to check the integrity of information transmitted over or stored in an unreliable medium is a prime necessity in the world of open computing and communications. Mechanisms that provide such integrity check based on a secret key are usually called "message authentication codes" (MAC). [11]

Authentication tag is used for providing some additional security to the SRTP packet. The optional MKI(master key identifier) and the RECOMMENDED authentication tags are the only fields defined by SRTP that are not in RTP [3]. So we have included this recommended portion of SRTP in our application which may be used to simultaneously verify both the data integrity and the authenticity of a message. The SHA-1 algorithm which we have used is a keyed-hashing algorithm involving a cryptographic hash function in combination with a secret key. So it accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC. Since the hash function used is SHA-1, so our resulting MAC can be termed as HMAC-SHA1.

HMAC-SHA1 is generated at the sender side first, and after the particular SRTP packet is received at the client side the hash code is calculated again, and it's verification takes place, and only after the successful matching of HMAC-SHA1 the data is considered to be authenticated. The HMAC-SHA1 value protects both the message's integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content. The difference between encryption and hashing is that the encrypted data can be decrypted, while hashed data cannot be decrypted because hashing data is a one-way process and there is no way to reverse the process.

5.7. Possible Attacks on Data and their Solution

The main feature to provide security against the theft of data is the Authorization process. Each user is given a password, and we believe that data theft can not take place till that person him/her self gives his/her password to an irresponsible person.

Another major attack can be the replay attack by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution. The Authorization process and Time stamping is the best way of preventing a replay attack, As our SRTP Header is providing a field of Time stamps itself, Sender sends the time on it's clock together with a MAC and by the check on the time stamps on each packet, this attack can be avoided easily. Receiver only accepts messages for which the timestamp is within a reasonable tolerance (Using this method Synchronization too is achieved). Also the sequence number of packets is part of this protection, they together provides protection against replay attacks. Authentication too is major security factor in this application which not only detects the security threats related to unauthorized replay but also the deletion, insertion and manipulation of messages. A mechanism has been incorporated in the application to drop adulterated packets.

SRTP uses encryption and authentication to minimize the risk of denial of service (DoS) attacks. The denial of Receipt attack prevention is tackled in a way that the receiver keeps sending a message after a fixed interval (10 seconds) and it verifies that he/she is receiving the data.

5.8. Client Leaving Notification

Furthermore, there is a need of keeping the record of the clients who are leaving the services of server, so here we have incorporated a SRTCP service of BYE packet. Client sends a BYE packet when it leaves. And for keeping the record of the clients who are still active, each client keeps sending “Alive” message after every 10 seconds. If for five consecutive turns, the server doesn't receive the “Alive” message then it is assumed that the client is dead. It is the case only when client leaves in an abnormal manner and of course on leaving the session in timely manner the client will send a BYE packet and the server would come to know that a particular client has left.

6. Results

In the era of fast multimedia, time is a major factor to be considered, everyone needs to do their work in fixed time interval. SVS application is providing fast delivery of data, all the techniques used in this application to process data are not taking much of the extra time, there is some micro seconds time difference between the data traveling without encryption processing and with encryption processing.

To examine these scenarios, we have used a Get Time() function in our application, which is keeping record of the time of every packet, but the difference between two consecutive packets is too short, so we have adopted another strategy which is measure time differences; we have recorded the time after every 100 packets, and then drawn the graphs to analyze the time differences which our application is causing due to it's encryption and key management techniques.

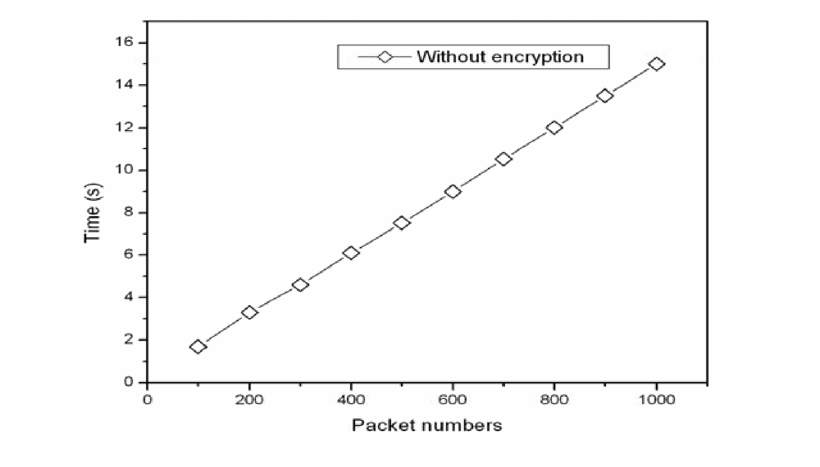
Two of the major graphs are made for analyzing this feature, As in this application compression is not our focus, so we are not considering any type of bandwidth utilization, our only concern is the time taken by our application after the implementation of it's security techniques upon data, two different scenarios are described as under:

6.1. First Scenario

In first scenario we plotted a graph (Figure 4) between time and packet numbers when data is traveling without any key management mechanism and encryption techniques. We have considered the first 1000 packets only, and get different time at different intervals (after every 100 packets) and the recorded initial and final time are 1206203783 seconds and 1206203798 seconds. For simplicity we have arranged this time in simple seconds varying from 0 seconds to 15 seconds almost.

So the graph between time and packet numbers is:

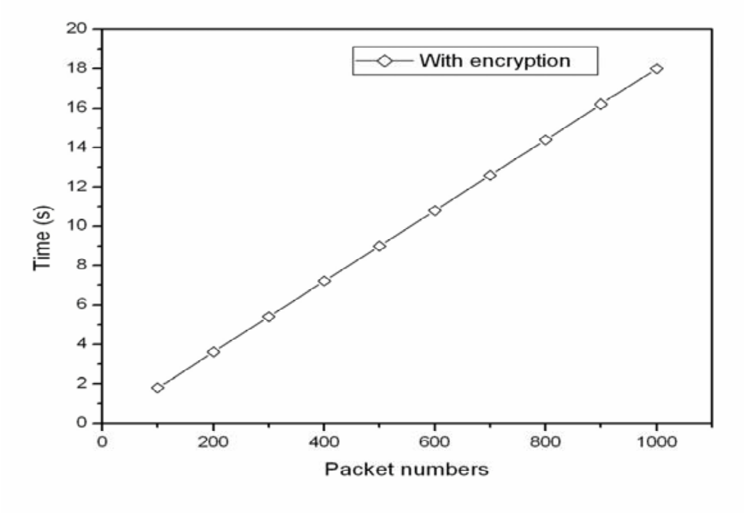
Figure 4: Plain video chunks traveling across the network in different time stamps



6.2. Second Scenario

In second scenario we plotted a graph (Figure 5) again between time and packet numbers when data is traveling after the key management and exchange mechanism and after being encrypted and embedded in SRTP packet. Here too, we considered the first 100 packets and by GetTime() function. We recorded their time intervals after every 100 packets, and the time we get is between 1206204204 and 1206204222 seconds. Again after simplification of seconds, this time varies from 0 to 18 seconds, the graphs is as under:

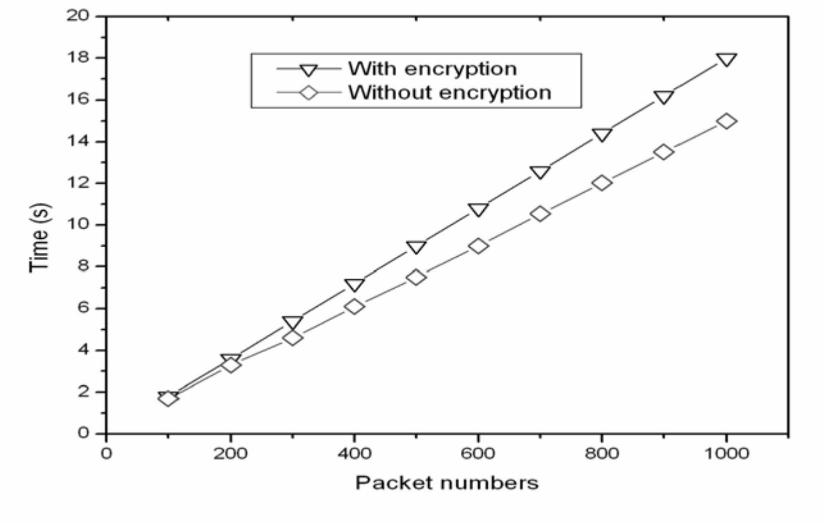
Figure 5: Encrypted SRTP video packets traveling across the network in different time stamps



6.3. Time Usage Comparison between Both Graphs

There is another graphs (Figure 6) which is clearly displaying the consumed time difference of both of the scenarios, and it is clear that our application had put only a three seconds time burden upon a 10 MB video file,

Figure 6: Comparison of both Scenarios



Our application is using block size of 4096 bytes for each data packet, so almost 266240 bytes are traveling upon the network every one second; this is making the data delivery faster.

7. Conclusion

The emphasis of this paper is to develop an environment with security infrastructure that performs secure multimedia streaming to users for the prevention of security threats. Different techniques, algorithms and protocols are put together in such a way that they are providing a best security solution to the data traveling upon the network, Also it is observed in this application that even after a great deal of data processing; the video and sound quality is same as in its original place. So this application has not merely made the secure and fast delivery of data but also it has maintained the picture and sound quality of streamed data.

We hope that this step towards provable security of real world implementations will be a motivating point for further research in the field of multimedia security. Real time video files can be made to travel over the network securely using the techniques described in this paper. The flexibility of this application further allows the implementation of other key management and encryption algorithms easily. What is described in this paper can be go a long way towards finding solutions to the problem of other security attacks on data transmitted over computer networks. Hopefully it will be a motivating point for further research in this area of security.

References

[1] E. Rescorla, Diffie-Hellman Key Agreement Method, Request for Comments: 2631, June 1999

[2] P. Chown, Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), Request for Comments: 3268, June 2002

[3] M. Baugher et. al., SRTP: The Secure Real-time Transport Protocol, Request for Comments: 3711, March 2004

[4] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, Request for Comments: 3550, July 2003

[5] J. Postel, User Datagram Protocol, Request for Comments: 768, 28 August 1980

[6] H. Schulzrinne, GMD Fokus, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, Request for Comments: 1889, January 1996

[7] Conversational video over IP BT Technology Journal • Vol 22 No 2 • April 2004

[8] (INTEROP LABS) Voice over IP Wireless and Security Lab -What Is SRTP? May 20-25, 2007

[9] National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Se.

[10] Schulzrine, h. and S. Casner, “RTP profile for Audio and Video Conferences with Minimal Control”, Request for Comments: 3551, July 2003.

[11] Hugo Krawczyk, Mihir Bellare, Ran Canetti, HMAC: Keyed-Hashing for Message Authentication Request for Comments 2104, February 1997.