Effective and Secure Scheme for Video Streaming Using SRTP

P. Iyyanar, M. Chitra, and P. Sabarinath

Abstract—**Real-time of live video or stored video is the predominant part of the real-time multimedia networking. In the streaming technology, the video file need not be downloaded in full, but is being played out while content of the video file are being received and decoded. To send video across the Internet has two approaches: video data compression and design of communication protocol. One approach is to design a low bit rate coder as well as protecting the result of the bit stream using UDP. Another approach is to design new transport protocol by using video coding algorithm. Security plays a major role in streaming application. In this paper, we present an efficient solution for encryption on the video streams using AES-CFB on un-trusted client–server networks. The valuation of the prototype system is to provide a safer transmission across the Internet with cost-effective and quality-guaranteed manner while using SRTP.**

Index Terms—**Video streaming, AES, RTP, RTSP, SRTP, SRTCP, UDP, H.264.**

I. Introduction

Streaming visual data to different users is becoming ever more 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 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.

II. Video Streaming

Streaming is the process of playing the audio and video file still it downloading. Video streaming [1] refers to the real time transmission of stored video or live video. There are two types for transmission of stored video across the Internet available. One is download mode and another one is streaming mode.

In the download mode, a user downloads the entire video file and then plays back the video file. In the steaming mode, the video file need not be downloaded in full, but is being played out while parts of the video file being received.

In real time nature, video streaming has bandwidth, delay and loss of packet requirements. In video streaming, raw video and audio data are compressed by video compression and audio compression algorithms and saved in storage devices. If the client gives the request, a streaming server retrieves compressed audio and video data from storage devices to that particular client. When start to send the audio and video streams across the network the transport protocol packetize the compressed bit streams and send the audio-video packets to the Internet. For packets that are successfully delivered to the receiver, they first pass through the transport layers and then are processed by the application layer being decoded at the audio- video decoder.

To improve the streaming quality while transmission of audio and video data, continuous media distribution services and media synchronization are developed in the Internet.

III. Video Compression Technique (H.264)

H.264 [2] is a business standard for video compression, the method of converting digital video into a format that takes up a smaller amount capacity when it is stored or transmitted. Video compression is a vital technology for applications such as digital television, Video, mobile TV, videoconferencing and internet video streaming. Customizing video compression makes it possible for products from different producers (e.g. encoders, decoders and storage media) to inter-operate. An encoder transfers video into a compressed format and a decoder convert's compressed video back into an uncompressed format.

IV. Real-Time Transport Protocol

Real-time transport protocol (RTP) [3] is an IP-based protocol providing support for the transport of real-time data such as video and audio streams. RTP offer end-to-end delivery services for data with real-time characteristics. The services provided by RTP consist of time reconstruction, loss detection, security and content identification.RTP itself on the other hand, does not provide all of the functionality required for the transport of data and, therefore, applications usually run it on top of a transport protocol such as UDP. RTP does not deal with resource reservation and does not assurance quality-of-service (QoS) for real-time services. It requests support from lower layers that actually have control over resources in switches and routers.

RTP has its companion RTP control Protocol (RTCP) [5], which is used to control the flow and quality of data and allow the recipient to send feedback to the sources between

Manuscript received June 26, 2012, revised September 01, 2012. This work will be supported in part of Ph.D work.

P. Iyyanar and M. Chitra are with the Department of Information Technology, Sona College of Technology, Salem, Tamilnadu, India. (e-mail: inr.info@gmail.com; drmc@sonatech.ac.in).

P. Sabarinath is with the Tata Elxsi, Chennai and Tamilnadu, India. (e-mail: sabarinath.pkk@gmail.com).

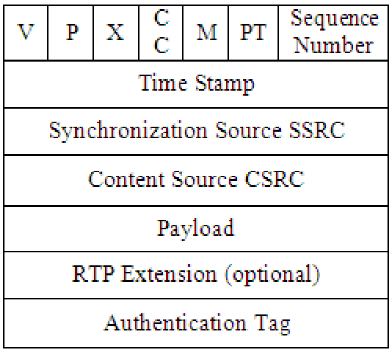
10.7763/IJMLC.2012.V2.252

client server systems. RTCP has five types messages those are sender report, receiver report, source description message, bye message and application specific message.

V. Secure Real-Time Transport Protocol

The Secure Real-time Protocol [4] is a profile of the Real-time Transport Protocol (RTP) offering not only confidentiality, but also message authentication, and replay protection for the RTP traffic as well as RTCP (Real-time Transport Control Protocol).

SRTP offers a structure for encryption and message authentication of RTP and RTCP streams. SRTP can achieve high throughput and low packet expansion.



V	P	X	C C	M	PT	Sequence Number
Time Stamp						
Synchronization Source SSRC						
Content Source CSRC						
Payload						
RTP Extension (optional)						
Authentication Tag						

Fig. 1. SRTP Packet format

SRTP is independent of a specific RTP stack implementation and of a specific key management standard, but Multimedia Internet Keying (MIKEY) has been designed to work with SRTP.

In comparison to the security options for RTP there are some advantages to using SRTP. The advantages over the RTP standard security and also over the H.264 security for media stream data are listed below

SRTP provides increased security, achieved by

- Confidentiality for RTP as well as for RTCP by encryption of the respective payloads.
- Integrity for the entire RTP and RTCP packets, together with replay protection.
- The possibility to refresh the session keys periodically, which limits the amount of cipher text produced by a fixed key, variable for an adversary to cryptanalysis.
- An extensible framework that permits upgrading with new cryptographic algorithms.
- A secure session key derivation with a pseudo-random function at both ends.
- The usage of salting keys to protect against precomputation attacks.
- Security for unicast and multicast RTP applications.

VI. Proposed System For Secured Video Streaming

In the secured video streaming implementation, the compressed audio-video data is retrieved and packetized at the SRTP layer for the Data plane at the sending side. The SRTP packetized streams provide timing and synchronization information and as well as sequence number. The SRTP packetized streams are then passed to the UDP layer and the IP layer. The resulting IP packets are transported across the Internet. At the receiver side the media streams are processed in the reversed manner before their presentation. For the control plane, SRTCP packets and RTSP packets are multiplexed at the UDP layer and are moved to the IP layer for transmission across the Internet.

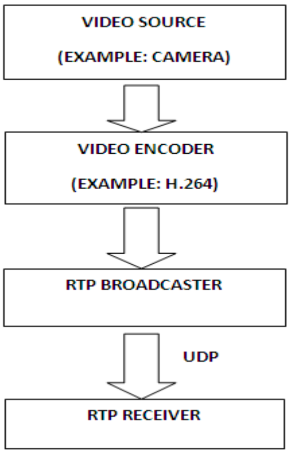


Fig. 2. Video Streaming across the Internet

Fig. 2 shows the Video Streaming across the Internet. In our system we will capture the video from camera or stored audio/ video files and have to encode the video by using H.264 then that video will be separate packet by packet then it will be sent as streams through RTP protocol and that will be received in another side of RTP media player.

A. *Architecture of Secured Video Steaming*

The big breakthrough that enables the streaming revolution is the adoption of a Internet protocol called the User Datagram Protocol (UDP) and new encoding techniques that compressed audio files into extremely small packets of data. UDP made streaming media possible by transmitting data more powerfully than previous protocols (HTTP and TCP) from the host server over the Internet to the client player or end listener. More recent protocols such as the Real-Time Streaming Protocol (RTSP) [6] are making the transmission of data even more efficient.

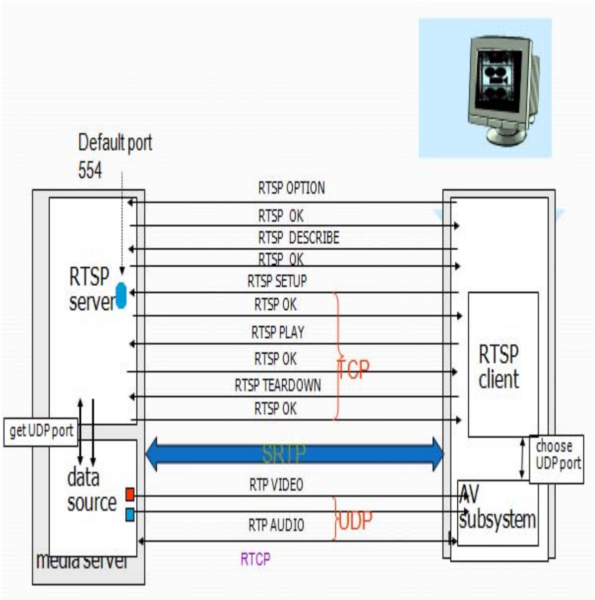


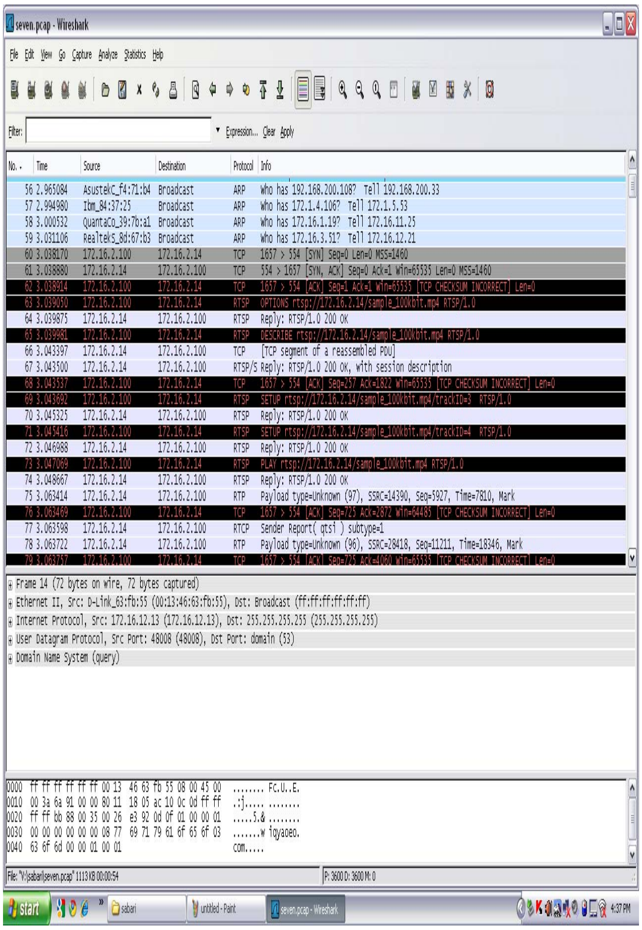
Fig. 3. Video streaming from server

Fig. 3 shows the RTSP client requests the audio and video files from the RTSP server. First we have to implement the RTSP client operation such as RTSP OPTION, RTSP DESCRIPE, RTSP SETUP, RTSP PLAY, RTSP TEARDOWN and RTSP OK for the remote control usage for end-user. Table 1 shows the methods in RTP.

TABLE I: RTSP METHODS

Methods	Description
Options	Get available methods
ANNOUNCE	Get description of media object
PLAY	Start playback, reposition
RECORD	Start recording
REDIRECT	Redirect client to new server
PAUSE	Halt delivery, but keep state
SET-PARAMETER	Device or encoding control
TEARDOWN	Remove state

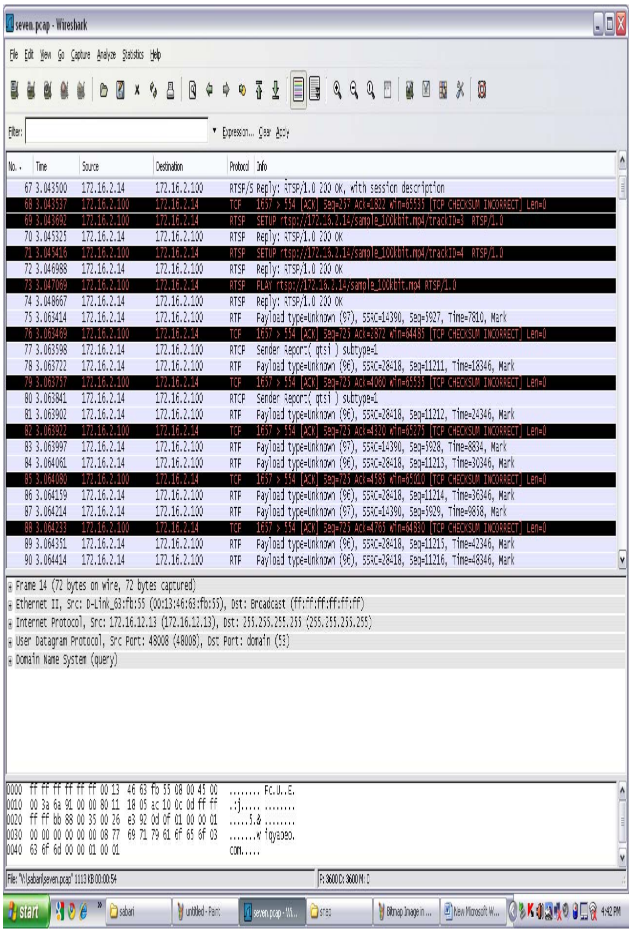
In the RTSP and RTP implementation we are using the Wireshark tools for simulation purpose in the client – server connection in which how the packets are received from server to clients. Fig. 5 and Fig. 6 describe the RTSP Client and Server operation.



```
No. Time       Source             Destination  Protocol Info
 56 2.965084   AsustekC_f4:71:b4  Broadcast    ARP    Who has 192.168.200.108?  Tell 192.168.200.33
 57 2.994980   Ibm_84:37:25       Broadcast    ARP    Who has 172.1.4.106?  Tell 172.1.5.53
 58 3.000532   QuantaCo_39:7b:a1  Broadcast    ARP    Who has 172.16.1.19?  Tell 172.16.11.25
 59 3.031106   RealtekS_8d:67:b3  Broadcast    ARP    Who has 172.16.3.51?  Tell 172.16.12.21
 60 3.038170   172.16.2.100       172.16.2.14  TCP    1657 > 554 [SYN] Seq=0 Len=0 MSS=1460
 61 3.038880   172.16.2.14        172.16.2.100 TCP    554 > 1657 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
 62 3.038914   172.16.2.100       172.16.2.14  TCP    1657 > 554 [ACK] Seq=1 Ack=1 Win=65535 [TCP CHECKSUM INCORRECT] Len=0
 63 3.039050   172.16.2.100       172.16.2.14  RTSP   OPTIONS rtsp://172.16.2.14/sample_100kbit.mp4 RTSP/1.0
 64 3.039875   172.16.2.14        172.16.2.100 RTSP   Reply: RTSP/1.0 200 OK
 65 3.039981   172.16.2.100       172.16.2.14  RTSP   DESCRIBE rtsp://172.16.2.14/sample_100kbit.mp4 RTSP/1.0
 66 3.043397   172.16.2.14        172.16.2.100 TCP    [TCP segment of a reassembled PDU]
 67 3.043500   172.16.2.14        172.16.2.100 RTSP/S Reply: RTSP/1.0 200 OK, with session description
 68 3.043537   172.16.2.100       172.16.2.14  TCP    1657 > 554 [ACK] Seq=257 Ack=1822 Win=65535 [TCP CHECKSUM INCORRECT] Len=0
 69 3.043692   172.16.2.100       172.16.2.14  RTSP   SETUP rtsp://172.16.2.14/sample_100kbit.mp4/trackID=3  RTSP/1.0
 70 3.045325   172.16.2.14        172.16.2.100 RTSP   Reply: RTSP/1.0 200 OK
 71 3.045416   172.16.2.100       172.16.2.14  RTSP   SETUP rtsp://172.16.2.14/sample_100kbit.mp4/trackID=4  RTSP/1.0
 72 3.046988   172.16.2.14        172.16.2.100 RTSP   Reply: RTSP/1.0 200 OK
 73 3.047069   172.16.2.100       172.16.2.14  RTSP   PLAY rtsp://172.16.2.14/sample_100kbit.mp4 RTSP/1.0
 74 3.048667   172.16.2.14        172.16.2.100 RTSP   Reply: RTSP/1.0 200 OK
 75 3.063414   172.16.2.14        172.16.2.100 RTP    Payload type=Unknown (97), SSRC=14390, Seq=5927, Time=7810, Mark
 76 3.063469   172.16.2.100       172.16.2.14  TCP    1657 > 554 [ACK] Seq=725 Ack=2872 Win=64485 [TCP CHECKSUM INCORRECT] Len=0
 77 3.063598   172.16.2.14        172.16.2.100 RTCP   Sender Report( qtsi ) subtype=1
 78 3.063722   172.16.2.14        172.16.2.100 RTP    Payload type=Unknown (96), SSRC=28418, Seq=11211, Time=18346, Mark
 79 3.063757   172.16.2.100       172.16.2.14  TCP    1657 > 554 [ACK] Seq=725 Ack=4060 Win=65535 [TCP CHECKSUM INCORRECT] Len=0

⊞ Frame 14 (72 bytes on wire, 72 bytes captured)
⊞ Ethernet II, Src: D-Link_63:fb:55 (00:13:46:63:fb:55), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol, Src: 172.16.12.13 (172.16.12.13), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: 48008 (48008), Dst Port: domain (53)
⊞ Domain Name System (query)

0000  ff ff ff ff ff ff 00 13  46 63 fb 55 08 00 45 00   ........ Fc.U..E.
0010  00 3a 6a 91 00 00 80 11  18 05 ac 10 0c 0d ff ff   .:j..... ........
0020  ff ff bb 88 00 35 00 26  e3 92 0d 0f 01 00 00 01   .....5.& ........
0030  00 00 00 00 00 00 08 77  69 71 79 61 6f 65 6f 03   .......w iqyaoeo.
0040  63 6f 6d 00 00 01 00 01                            com.....

File: "V:\sabari\seven.pcap" 1113 KB 00:00:54    P: 3600 D: 3600 M: 0
```

Fig. 4. RTSP client connection

The audio and video is compressed and send to the transport layer RTP/UDP then it will go to the IP layer and then transmitted to the Internet. While transmit the audio-video packet to the transport layer the SRTP protocol is used for the packet transaction.



```
No. Time       Source        Destination  Protocol Info
 67 3.043500   172.16.2.14   172.16.2.100 RTSP/S Reply: RTSP/1.0 200 OK, with session description
 68 3.043537   172.16.2.100  172.16.2.14  TCP    1657 > 554 [ACK] Seq=257 Ack=1822 Win=65535 [TCP CHECKSUM INCORRECT] Len=0
 69 3.043692   172.16.2.100  172.16.2.14  RTSP   SETUP rtsp://172.16.2.14/sample_100kbit.mp4/trackID=3  RTSP/1.0
 70 3.045325   172.16.2.14   172.16.2.100 RTSP   Reply: RTSP/1.0 200 OK
 71 3.045416   172.16.2.100  172.16.2.14  RTSP   SETUP rtsp://172.16.2.14/sample_100kbit.mp4/trackID=4  RTSP/1.0
 72 3.046988   172.16.2.14   172.16.2.100 RTSP   Reply: RTSP/1.0 200 OK
 73 3.047069   172.16.2.100  172.16.2.14  RTSP   PLAY rtsp://172.16.2.14/sample_100kbit.mp4 RTSP/1.0
 74 3.048667   172.16.2.14   172.16.2.100 RTSP   Reply: RTSP/1.0 200 OK
 75 3.063414   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (97), SSRC=14390, Seq=5927, Time=7810, Mark
 76 3.063469   172.16.2.100  172.16.2.14  TCP    1657 > 554 [ACK] Seq=725 Ack=2872 Win=64485 [TCP CHECKSUM INCORRECT] Len=0
 77 3.063598   172.16.2.14   172.16.2.100 RTCP   Sender Report( qtsi ) subtype=1
 78 3.063722   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (96), SSRC=28418, Seq=11211, Time=18346, Mark
 79 3.063757   172.16.2.100  172.16.2.14  TCP    1657 > 554 [ACK] Seq=725 Ack=4060 Win=65535 [TCP CHECKSUM INCORRECT] Len=0
 80 3.063841   172.16.2.14   172.16.2.100 RTCP   Sender Report( qtsi ) subtype=1
 81 3.063902   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (96), SSRC=28418, Seq=11212, Time=24346, Mark
 82 3.063922   172.16.2.100  172.16.2.14  TCP    1657 > 554 [ACK] Seq=725 Ack=4320 Win=65275 [TCP CHECKSUM INCORRECT] Len=0
 83 3.063997   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (97), SSRC=14390, Seq=5928, Time=8834, Mark
 84 3.064061   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (96), SSRC=28418, Seq=11213, Time=30346, Mark
 85 3.064080   172.16.2.100  172.16.2.14  TCP    1657 > 554 [ACK] Seq=725 Ack=4585 Win=65010 [TCP CHECKSUM INCORRECT] Len=0
 86 3.064159   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (96), SSRC=28418, Seq=11214, Time=36346, Mark
 87 3.064214   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (97), SSRC=14390, Seq=5929, Time=9858, Mark
 88 3.064233   172.16.2.100  172.16.2.14  TCP    1657 > 554 [ACK] Seq=725 Ack=4765 Win=64830 [TCP CHECKSUM INCORRECT] Len=0
 89 3.064351   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (96), SSRC=28418, Seq=11215, Time=42346, Mark
 90 3.064414   172.16.2.14   172.16.2.100 RTP    Payload type=Unknown (96), SSRC=28418, Seq=11216, Time=48346, Mark

⊞ Frame 14 (72 bytes on wire, 72 bytes captured)
⊞ Ethernet II, Src: D-Link_63:fb:55 (00:13:46:63:fb:55), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol, Src: 172.16.12.13 (172.16.12.13), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: 48008 (48008), Dst Port: domain (53)
⊞ Domain Name System (query)

0000  ff ff ff ff ff ff 00 13  46 63 fb 55 08 00 45 00   ........ Fc.U..E.
0010  00 3a 6a 91 00 00 80 11  18 05 ac 10 0c 0d ff ff   .:j..... ........
0020  ff ff bb 88 00 35 00 26  e3 92 0d 0f 01 00 00 01   .....5.& ........
0030  00 00 00 00 00 00 08 77  69 71 79 61 6f 65 6f 03   .......w iqyaoeo.
0040  63 6f 6d 00 00 01 00 01                            com.....

File: "V:\sabari\seven.pcap" 1113 KB 00:00:54    P: 3600 D: 3600 M: 0
```

Fig. 5. RTSP server

B. *Packet Transmission*

There is a number of Internet drafts describe SRTP packetization schemes for MPEG-4 video data. Media aware packetization is a principle in SRTP, so it is likely that several SRTP scheme will be needed to suit the different kind of media, audio, video and so forth. The SRTP time stamp corresponds to the presentation time if the earliest access unit is within the packet. SRTP packets have sequence numbers in transmission order. The payloads logically or physically have synchronization layer sequence numbers, which are decoding order, for each elementary stream.

The MPEG-4 time scale is the time-stamp resolution in the case of MPEG-4 systems and must be used as the SRTP time scale. Streams should be synchronized using RTP techniques which is RTCP sender report. When the MPEG-4 object clock reference is used, it is logically mapped to the network time protocol time axis used in RTCP.

C. *AES Chiper*

The Rijndael proposed for AES [7] defined a cipher in which the block length and key length can be independently specified to be 128,192 or 256 bits. The AES requirement uses the same three key size alternatives but limits the block length to 128 bits.

Stream cipher techniques have been used in video streaming over the multimedia network. A stream cipher is a symmetric encryption algorithm in which cipher text output is produced bit-by-bit or byte-by-byte from a stream of plaintext input. A block cipher algorithm is used for providing the data security.

AES Cipher Feedback (CFB) [8] algorithm is used for general-purpose stream-oriented transmission and provides the authentication. A stream cipher eliminates the need to pad a message to be an integral number of blocks. This is used to operate in real time. In this method a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher.

In the encryption side, the input is a b-bit shift register that is initially set to some initialization vector (IV). The leftmost s bits of the output of the encryption function are XORed with the first segment of plaintext P1 to produce the first unit of cipher text C1, which is then transmitted.

$$Ci = Ek(Ci-1) \oplus Pi \tag{1}$$

$$Pi = Ek(Ci-1) \oplus Ci \tag{2}$$

$$C0 = IV \tag{3}$$

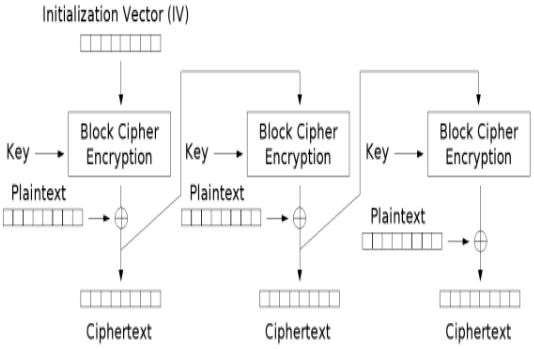


Fig. 6. Cipher Feedback mode encryption

In addition, the contents of the shift register are shifted left by s bits and C1 is placed in the rightmost s bits of the shift register. This process continuous until all the plaintext units has been encrypted.

D. Encryption in SRTP Packet

In our proposed system, we have to modify an encryption method in the SRTP packet format at RTP payload field. In the related work of SRTP implementation, AES – CBC encryption algorithm and AES –CM algorithm were used. In our proposed system, we will apply AES – CFB algorithm for video streams encryption and decryption. The disadvantage of previous algorithm is while encrypting the video file it had encrypted block by block and it gave the general purpose block oriented transmission. In AES –CFB, the stream cipher will be encrypted bit by bit or byte by byte. The advantage of AES – CFB algorithm is to send the stream oriented data transmission and it will give more security than CBC and CM mode in Advanced Encryption Standard.

VII. Application Layer QoS

The purpose of application layer QoS control is to avoid congestion and maximize video quality in the presence of packet loss. The application layer QoS control techniques include congestion control and error control. These technique are employed by the end systems and do not require any QoS support from the network. For streaming video, congestion control obtains the form of rate control. There are three kind of rate control: source-based, receiver-based and hybrid rate control.

The source-based rate control is suitable for Unicast video and other two rate control for multicast video.

In unicast video streaming, the model based approach is based on a throughput of a UDP connection. Specifically, the throughput of a UDP connection can be characterized by the following equation:

$$\mu = (1.22 \text{ MTU}) / (\text{RTT} \sqrt{p}) \tag{4}$$

where μ is a throughput of a UDP connection. MTU is the packet size used by the connection,

RTT is the round-trip time for the connection and p is the packet-loss ratio experienced by the connection. This equation is used to determine the sending rate of the video stream. Thus, the video connection could avoid congestion in a way similar to that of UDP and it can compete fairly with UDP flows.

VIII. The Valuation of Secure Video Streaming

In our proposed system, it will provide secure delivery of video data transmission. The time difference between without security of video data transmission and with security of video data transmission will be few seconds. Due to encryption techniques in our system we can find the difference between secure and non secure video streaming transmission. The secured video streaming file will take more time (seconds) than non secure video streaming, but it will provide safer video data transmission in un-trusted client – server networks.

IX. Conclusion

RTP server applications transmit captured or stored media streams across the network. The main challenge in designing a video streaming application across the multimedia networks is how to deliver video streams to users with minimal replay jitters with video data security and efficient video data transmission.

The media streams might be encoded in multiple media formats and sent out on several RTP sessions for conferencing with heterogeneous receivers. This paper proposed a framework for video streaming services using SRTP through the client-server network.

References

[1] Lei Chen, Chung- wei Lee "Multi-level secure video streaming over SRTP," *Proceedings of the 43rd ACM Southwest Conference, Kennesaw, GA, USA*, March 18-20, 2005.

[2] I. Richardson, An overview of H.264 Advanced Video, [Online]. Available: http://www.vcodex.com,2007.

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacob-son, "RTP: A Transport Protocol for Real-Time Ap-plications," *IETF RFC 3550*, 2003.

[4] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The Secure Real-time Transport Proto-col (SRTP)," *RFC 3711*, 2004

[5] C. Perkins, *RTP: Audio and Video for the Internet*, Ad-dison Wesley, 2003.

[6] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," *IETF RFC 2326* (proposed standard), Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2326.txt

[7] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," February 2002.
[8] William Stalling, "Cryptography and network Security" fourth Edition, 2006.

P. Iyyanar was born in Salem district of Tamilnadu, India. He is doing PhD (CSE) in Anna University of Technology, Coimbatore, India. He received ME-CSE degree from Anna University of Technology, Coimbatore in 2009 and B.E-IT degree from Periyar University, Salem, India in 2003. His area of Research is Multimedia Networking. Currently he is working as a Assistant Professor in Department of IT, Sona College of Technology, Salem, India. He has work experience of 8 years in teaching. He is a life time member of ISTE and IACSIT.

Dr. M. Chitra was born in Namakkal district of Tamilnadu, India. She received her B.Sc. degree in Mathematics from the University of Madras Chennai in 1993, M.Sc. in Mathematics from the Bharathidasan University Tiruchirappalli in 1995, and M.Phil.. in Mathematics from Avinashilingam University Coimbatore in 1997, M.S (IT) from the Bharathidasan University Tiruchirappalli in 2002 and PhD in Mobile Networks from Anna University Chennai India in 2009 Her research interest includes Computer Networks, Mobile Networks and AdHoc Mobile Networks. She has four international publications to her credit. Currently she is working as Professor in the Department of Information Technology at Sona College of Technology, Salem, India. She is an approved supervisor of Anna University Coimbatore for PhD programme and presently guiding 8 Research Scholars. She is a life time member in ISTE computer Society of India.

P. Sabarinath is working as a Specialist in TATA ELXSI, Chennai. He received B.E-ECE degree from Periyar University, Salem, India in 2003. He has worked in various industries His area of Research is Multimedia Networking and Image processing in which he has published an IEEE paper. He is currently working in wireless domain. He has a work experience of 8 years in both hardware and software industries.