Performance Comparison of Encryption Techniques for MPEG2-TS Video Streaming

Poonam Yadav
Dept.VLSI & EMBEDDED SYSTEM
Ganpat University
Mehsana, India

***Abstract*—Now a day IPTV, VideoOnDemand are using MPEG2-TS Video streaming and security of the multimedia data is playing a vital role in it .In this paper the performance of AES is compared to DES and blowfish techniques which is proving efficient technique simulated on different data types (audio, video, single program MPEG2-TS) between peer to peer communication using UDP transmitter/Receiver**

Keywords—MPEG2-TS,Video Streamimg,Encryption,AES.

I. INTRODUCTION

MPEG2-TS Technology

MPEG transport stream (MPEG-TS, MTS or TS) is standard format for transmission and storage of audio, video, Program and System Information Protocol (PSIP) data. It is used in broadcast systems such as DVB, ATSC and IPTV. MPEG-2 Transport Streams are composed of 188 bytes TS Packets, each with a 4byte header. Some TS packets contain an optional Adaptation Field whose size depends on flags set in
the packet header and which may contain timing information, pad bytes, and other data. TS packet payloads may contain program information as well as Packetized Elementary Streams (PES), typically video and audio streams.

Video Streaming

Video streaming can be delivered in complete video package of linear programming, as a subscription services or as pay-per-view. Video Streaming application are like Internet broadcasting (corporate communication), education (web lectures &distance learning), web based channel video on demand and internet and intranet browser of content (asset management).Such system needed security precaution for networked multimedia application.

Time delay plays an important role while playing a video stream over a network in real time. Video frame need to be displayed at certain rate, therefore sending and receiving encrypted packets must be sent in minimum amount of time to utilize the admissible delay. Whenever the receiver asks for the video, VideoOnDemand should play the stream. It is called real time streaming as no playback or buffer is used in playing video stream. Challenges that came across multimedia security

1) Even though MPEG2TS is compressed video size is large.
2) Video on demand like multimedia need to be run in real time.
3) Processing multimedia stream should be having a bounded Performance.
4) Encryption techniques should give minimum overhead in Processing.

The goal of this research focused on the following points: First, implementing AES for MPEG2TS in real time video transmitting system. Second comparing performance of the AES with respect to two techniques over peer to peer network. Third evaluation of overhead resulting from multimedia (text, audio, video, single program MPEG2TS) due to three techniques (AES, DES, Blowfish).

This paper is organized as follows. In section 2, the concept of encryption techniques is given. Then in section 3 a Module implementation and its brief description is discussed. In section 4 performance analysis parameter. In section 5 result of using. Finally the conclusion is drawn in section 6.

II. BASIC CONCEPTS OF VIDEO ENCRYPTION

The two techniques used for encryption and decryption of a plaintext or a video stream are public key encryption and secret key encryption. For real time video conferencing public key cryptography is not applicable for security, as its operations take more time, which is not suitable for video conferencing.

A video streaming is made up of seven building blocks, as illustrated in figure1.Raw audio and video data are pre-compressed by audio and video compression algorithm and saved in storage devices. Server retrieves compressed audio/video data from storage devices on request of client. And then the QoS control module adapts the video/audio bit-streams according to the network status and QoS requiremens.Transport protocol then packetized there compressed bit-streams and send packet frame to Internet IP networks. To keep eye on packets loss and packets damaged, continuous media distribution services are deployed in the internet. At the receiver side the successfully receive packets are processed by application layer before being decoded at the

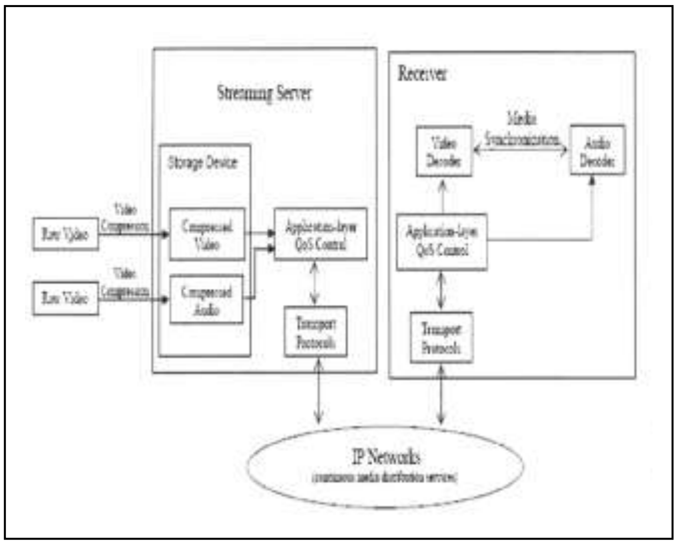

Fig.1One way data flow block diagram for captured multimedia devices

Decoder. Synchronization mechanism is required to achieve synchronization between video and audio presentations [5].
There are many video encryption algorithms. Such encryption technique can be classified as follows: naive algorithm, selective algorithm, Zigzag algorithm, RC4 and AES [6].A review of each one is briefly given. The idea of naïve encryption [3] is to deal with the video streams as text data. A review of three techniques used in this paper is given.
DES is the archetypal block cipher, an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits.

Blowfish has a 64-bit block size and a variable key length from 32 bits up to 448 bit. Each line represents 32 bits. The algorithm keeps two sub key arrays: the 18-entry P-array and four 256-entry S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries. The function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 2^{32} and XORed to produce the final32-bit output.

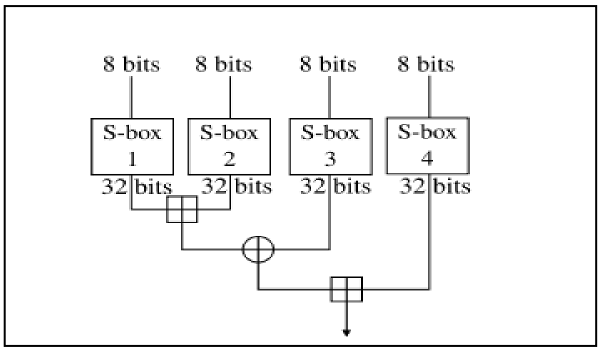

Fig. 2 block diagram of Blowfish[9]

The AES algorithm is essentially Rijndael [7] symmetric key cryptosystem that processes 128-bit data blocks using cipher keys with lengths of 128, 192, or 256 bits. Rijndael is more scalable and can handle different key sizes and data block sizes, however they are not included in the standard. Also the basic blocks of AES operation is shown in figure 3.Further details about this algorithm can be found in [8].

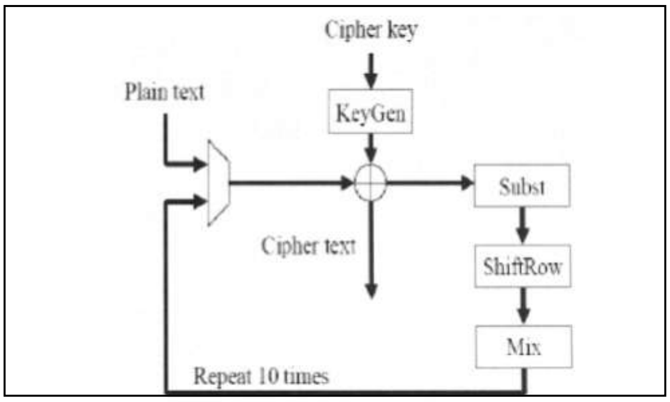

Fig. 3 block diagram of AES [8]

III. MODULE IMPLEMENTATION AND ITS BRIEF DESCRIPTION

The modules used in the implementation are as follows:

A. *UDP Transmitter/Receiver*

UDP Protocol is used for transmitting the stream of MPEG2-TS in form of packet on the network for peer to peer transmissions.

B. *Buffer Manager Module*

It provide a temporary storage to the stream at the receiver and sender side so as to do continuous encryption and decryption of the input stream without affecting the video/audio to get played at the client side

C. *Encryption module*

It is implemented using openssl library and EVP interface function. The Openssl Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the Openssl toolkit and its related documentation.

D. *Encryption Function*

. Step1: Read MPEG2-TS file each 188 byte data in the buffer

Step2: size of output buffer for encryption should be 188+ EVP_MAX_IV_LENGTH.

Step3: Generate Key and Initialization vector using EVP_BytesToKey

Step4: Perform encryption using EVP_EncryptUpdate and EVP_EncryptFinal_ex on buffer data

Step5: Store the length of cipher text given in

EVP_EncryptFinal_ex output length for decryption

E. *Decryption Function*

Step1: Read MPEG2-TS file each 188 byte data in the buffer

Step2: Size of input buffer for encryption should be 188+ EVP_MAX_IV_LENGTH.

Step3: Generate Key and Initialization vector using EVP_BytesToKey of 128bit size

Step4: perform decryption using function call EVP_DecryptUpdate and EVP_DecryptFinal_ex on Buffer data

Step5: Store the length of cipher text given in EVP_DecryptFinal_ex output length equal to Plaintext.

IV. PERFORMANCE ANALYSIS PARAMETER

Te=encryption time to encrypt one block of 188 byte of MPEG2-TS data
Td=decryption time to decrypt one block of 188 byte of MPEG2-TS data
Performance analysis is done on the three inputs
1) MPEG2-TS Single program video (AV file) of 40000packet of 188byte each
2) MPEG2-TS only audio file of 3314packet of 188byte each
3) MPEG2-TS only video file of 36690 packet of 188byte each

At the receiver side playing the video and performance of each encryption techniques can be get by the difference between the original size and output file size of the media file and how efficiently it is playing on player.

V. RESULT

I have used ubuntu 12.04 machines. I have made final code by making a library having both encryption,decryption,buffer read, buffer write,UDP sender,UDP receiver function and test application will send input file from UDP sender by encrypting it to the client side where the receiver will decrypt it and play on the other machine.

So as to analysis the delay performance of encryption function and decryption I have calculate the timing parameter using clock function with different technique and it result are as follow..

Result on simulation on Input file MPEG-TS single program (AV file) original size=6.9MB:

Algorithm	Time to Encrypt(ms)(Te)	Time to Decrypt (ms)(Td)	Output file size(MB)	Performance cost
Blowfish	0.01	0.01	4.9	medium
DES	0.01	0.01	4.4	low
AES	0.01	0.01	5.1	high

Result on simulation on Input file MPEG-TS audio file original size=646 kb:

Algorithm	Time to Encrypt(ms)(Te)	Time to Decrypt (ms)(Td)	Output file size(kb)	Performance cost
Blowfish	0.00	0.00	561.2	medium
DES	0.00	0.00	503.8	low
AES	0.00	0.00	600	high

Result on simulation on Input file MPEG-TS video file original size=8.4MB:

Algorithm	Time to Encrypt(ms)(Te)	Time to Decrypt (ms)(Td)	Output file size(MB)	Performance cost
Blowfish	0.01	0.01	6.2	medium
DES	0.01	0.01	6	low
AES	0.01	0.01	6.8	high

CONCLUSION

Our study showed that the AES encryption algorithm can be used effectively to encrypt MPEG2-TS.The performance of AES encryption is sufficient to receive frames on time. It shows better performance than other two Techniques. Therefore we conclude that using AES encrypting MPEG2-TS Program is feasible.

REFERENCES

1. IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on Authentication, Copyright Protection, and Information Hiding, Vol. 13, No.8, August 2003.
2. X.Liu and A.M. Eskicioglu, "Selective Encryption of Multimedia Content in Distribution Networks: Challenges and New Directions" lASTED International Conference on Communications, Internet and Information Technology (CIIT 2003), Scottsdale, AZ, November 17-19, 2003.
3. D. R. Stinson, "Cryptography Theory and Practice," CRC Press, Inc., 2002.
4. William Stallings, "Cryptography and Network Security, Principles and Practice", Pearson education, Third Edition, 2005.
5. Chun-Shien L, "Multimedia Security Steganography and Digital Watermarking Techniques for Protection of Intellectual Property", Idea Group Publishing 2005.
6. T. Seidel, D. Socek, and M. Sramka, "Cryptanalysis of Video Encryption Algorithms", Proceedings of the 3rd Central European Conference on Cryptology TATRACRYPT 2003.
7. I. Agi and L. Gong, Empirical Study of Mpeg Video Transmissions," In
Proceedings of the Internet Society Symposium on Network and Distributed System Security, pages 137-144, San Diego, CA, February 1996.
8. Y. Li, Z. Chen, S. Tan, and R. Campbell, "Security enhanced mpeg player", In Proceedings of IEEE First International Workshop on Multimedia Software Development (MMSD'96), Berlin, Germany, March 1996.
9. Link: Wikipedia.org