On-Demand Secure Streaming of Multimedia Data over Cloud

Govinda .K [#1],Pavan Kumar Abburu [*2],Gangi Prathap Reddy [#3]
[#] SCSE, VIT University
Vellore, India
[1] kgovinda@vit.ac.in
[3] gangi.prathap@gmail.com
[*] SCSE, VIT University
Vellore, India
[2] spavanchowdary198@gmail.com

***Abstract*—On-demand digital communications on internet/cloud has been predominantly increasing recently. Due to the increase in the communication levels so did the security for the digital communications (called as multimedia Security) came into the system. Most Reliable and Secure communication is needed for most of the real time applications like Video conferencing, Medical Image Systems and Entertainment Services. The existing cryptographic techniques that we are currently using for the Data security will not be appropriate for the digital multimedia communication. Hence here comes an urgency to develop some new security protocols for our most rapidly developing Multimedia or Digital communications and to provide security and privacy for multimedia data. This paper provides a light weight mechanism for securing video file on-demand over Cloud.**

Keyword- Cloud TV, AES, Stream, Frame Set, Service *Provider.*

I. INTRODUCTION

Current Multimedia systems (digital communications) will make continuous media stream. It is very essential to stop the Hackers and eavesdroppers technically called as potential threats from corrupting or stealing the valuable information that is being passed through the Communication. Due to the demand for the streaming videos the applications for the streaming are going to be endless.

Main Aim Of Secured Video transmission is to provide : authentication, content tracking, conditional access, copy control, confidentiality. The level of Security differs from each and every Video application. The application that needed security can be classified as : Entertainment applications like VoD and pay TV. Personalized applications of video are like Business meetings, telemedicine, and diplomat dialogues etc. Though both the categories of applications need security, applications on Entertainment need low security when compared to the Video services that are personalized. Applications related to entertainment need to provide the video with more quality and timeliness. The Price of a High quality video is more when compared to that of the low quality videos, which makes the user to go for the high quality version of the video [7]. For all the entertainment video applications verification of the User identities is must, where as in personalized applications digital signs and certifications are used instead.

The industrial standard in video processing known as MPEG [1]. Providing security for many Multimedia applications like Video on-Demand service, broadcasting a video, video-conferencing and multimedia mails is must. A secured video transmission ensures the user such that no unapproved eavesdroppers can get the information from the video while it's being sent to receiver i.e the users those who paid for these services can only watch the videos and movies. If the video is more redundant it helps the attacker to easily rebuild the original video file. Data such as text and program code has less redundancy when compared to videos in its structure. All these factors make providing security for a MPEG video more challenging. Providing security for these MPEG video transmission involves in encrypting parts of the MPEG bit stream or the entire bit stream.

Applying cryptography algorithms such as Advanced Encryption Standard (AES) [2], International Data Encryption Algorithm (IDEA), and Data Encryption Standard (DES) is one of the ways for providing security to multimedia applications, though they involve in complex computations. When we consider the whole data we perform a heavy weight encryption algorithm which increases the problem and also latency. Whereas a light weight algorithm that provide enough security and also have a reasonable computation price for MPEG applications are used for encrypting the selective or partial data. In this paper, we present the most effective algorithm for MPEG video encryption which is based on AES encryption algorithm which is a light-weight encryption algorithm that uses selective encryption method. This algorithm is based on a light-weight selective encryption algorithm RVEA [3] which was based on DES and IDEA. By Adopting the AES algorithm to encrypt the data security had increased significantly. This Algorithm also reduced the computational time by reducing the maximum size of the bits selected.

Streaming can actually be done as a complete video package of a linear programming, as a subscription package (monthly pay), or as a pay per view service(pay based on data streamed). It might be as a part of a website or it be a tool for video preview and film dailies. Some applications are Internet broadcasting (corporate communications) education (viewing lectures and distance learning), web based channels (IP-TV, Internet radio), Video-on-demand (VOD) and Internet and intranet browsing of content (asset management). Such systems use different types of encryption techniques to increase the security precautions for networked multimedia applications. We are going to provide the security using AES algorithm.

Cloud Computing, being the current trend provides satisfaction to its customers with its on demand services and data of all kinds. Thus, when it comes for data available in the form of videos Cloud allows its customer not only to access videos that are on demand but also application in the form of services to view and manipulate it. The power of cloud lies in the fact that it runs services on the user machine without respect to the user system. We can see that now a day's most of the web based channels are concerning themselves to the Cloud because of its efficiency. The services like WeVideo and Cloud TV are very good examples of video streaming over cloud. To secure multimedia data over the cloud this paper proposed a secure streaming over the cloud using AES.

II. Literature Review

Currently we are using many encryption algorithms that are based on DES and IDEA in order to provide security for the MPEG video transmission. Coming to the security, Size of the resulting stream, and speed both these methods have their own Pros and Cons. Selective Algorithm, Naïve algorithm, Video Encryption Algorithm (VEA) [4], and Zig Zag permutation algorithm are some existing methods.

One straight-forward technique is encrypting the entire MPEG video stream using DES which is a standard encryption method. This method of encrypting is called as a Naïve algorithm approach. In the Naïve algorithm the MPEG bit-stream is considered as a text data and no special MPEG structures are used. This algorithm is more secured when compared to others but the computation time is very high so this algorithm is very slow. Just like the other standard encryption algorithm the bit stream size is not changed.

Using the basic features of the layered structure of MPEG files many selective algorithms were proposed. Since we know that P and B frames are more dependent on corresponding I frames, the basic selective algorithm is used for encrypting only I frames [5]. Most of the P and B frames carries the intra-coded I blocks so most part of the video may be visible to the users even after encrypting the I frames. Only 30-50% of the time is saved and size of the stream does not change even after encrypting I frames. Another selection scheme is proposed to encrypt only the headers of MPEG video. But this technique is not that effective because mostly the standard and basic information is stored in the headers and the index of each frame is well known because in order to perform the synchronization the video is indexed by the frames.

Zig-Zag permutation algorithm [6] provides a method in which the encryption is a part of the compression process of MPEG file. It uses a list of randomly chosen permutations to map each 8x8 block with a 1x64 vector instead zig-zag mapping of 8x8 block with a 1x64 vector. The speed of this algorithm is very fast and same when compared to the MPEG computational (encoding/decoding) time and also the size of the encrypted stream of MPEG increases significantly. But this algorithm can't with stand the plain text attack and cipher text attack.

In the Video Encryption Algorithm (VEA) a secret key is used to change all the sign bits randomly for all the DCT coefficients available for the MPEG video. Real-time VEA (RVEA) is the extension to VEA which encrypts the selected sign bits only and RVEA and VEA use DES/IDEA and XOR operations respectively. There is increase in the security for RVEA when compared to VEA since it adopts algorithms that uses secret key cryptography for data encryption. This Algorithm also reduced the computational time by reducing the maximum size of the bits selected. The Byte stream is simply scrambled using permutation in a Pure Permutation algorithm. The permutation key cardinality can be changed and it depends upon the level of security and requirements for the application. Here comes a problem with the Pure Permutation algorithm which can't handle the plaintext attacks but the Byte wise operations are very fast. Stream size is not increased unless we alter the key for each and every frame.

III. Proposed Method

In Our proposed model we applied the A light weighted encryption algorithm for a real time video streaming. We attempt to select specific frames (I frames) to encrypt. Where the encrypted video streams are the combinations of I, P, and B frames. We use AES algorithm to Encrypt and Decrypt the video by considering an input of 128 bits size both data and key each time. The key is exchanged between the user and the service provider using the DH Algorithm.

The basic process that involves in this work starts from considering an Input Video File from the User as an Input File, Which will be divided into frames. Basically now the Whole Video is converted into a series of images or frames. We are using a Selective encryption algorithm so we select I frames and we encrypt them using AES algorithm.

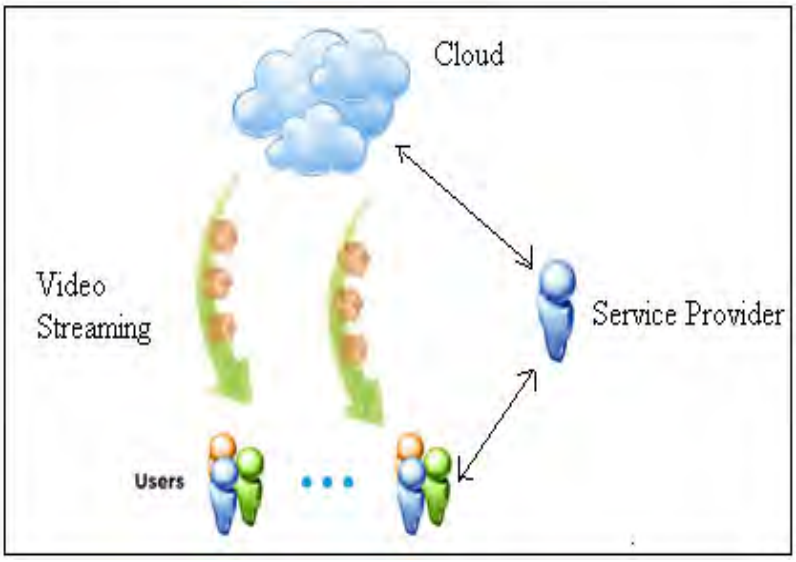

Fig 1. Streaming over Cloud

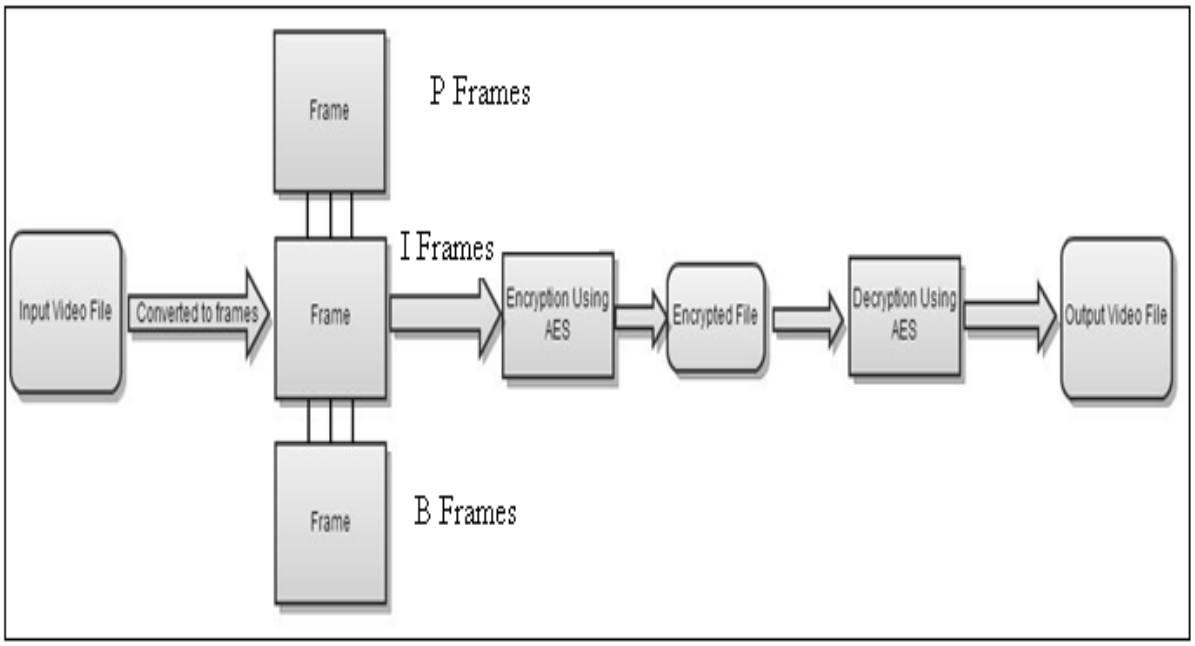

Fig 2. Detailed architecture

The basic steps that involves in our system are initially we divide the given video into Frames and then I frames are encrypted using AES as shown in Fig2. The encrypted stream is transmitted to user. The video is opened until you decrypt it. In our system by giving the same password which is exchanged with service provider. The Whole system architecture can be seen in the Fig2.

A. AES algorithm

Since AES is a Symmetric block cipher, we use the same key for both encryption and decryption. The Frames are considered as a one dimensional array of size 128 bits (The block length can be chosen independently from 128, 160, 192, 224, 256 bits but we used 128 bit length). In an AES algorithm both the key size and the block size should be same (since the size of the block is 128 bits the key length should also be a 128 bit).

The length of the key considered defines the number of AES parameters. Since we are considering a Key of length 128 bits we are going to use 10 rounds in our algorithm. It would be 12 and 14 rounds for 192 and 256 bits Key sizes respectively. Currently the key size of 128 bits is being used mostly.

AES algorithm is mainly used to provide these characteristics:

1. Resistance against all known attacks.
2. Speed and code compactness on a wide range of platforms.
3. Design Simplicity.

The Input is a 128 bits block and the key is of size 128 bits. The block data is sent into a State array which is altered at each and every stage of the algorithm. Both the 128 bits State array and key are depicted into a matrix of size 4x4 so total 16 Bytes. The Basic Operation process and steps involved in a AES algorithm are shown in the Figure3.

Algorithm starts with an Add Round Key method which will be followed by 9 rounds of 4 Basic operations

1. Substitute Bytes.
2. Shift Rows.
3. Mix Columns.
4. Add Round Key.

The 10^{th} round just leaves the Mix Columns Operation. Similarly in the decryption process in the 1^{st} nine rounds 4 basic operations are used.

1. Inverse Shift rows.
2. Inverse Substitute bytes.
3. Inverse Add Round Key.
4. Inverse Mix Columns.

The 10^{th} round leaves out the Inverse Mix Columns stage [2].

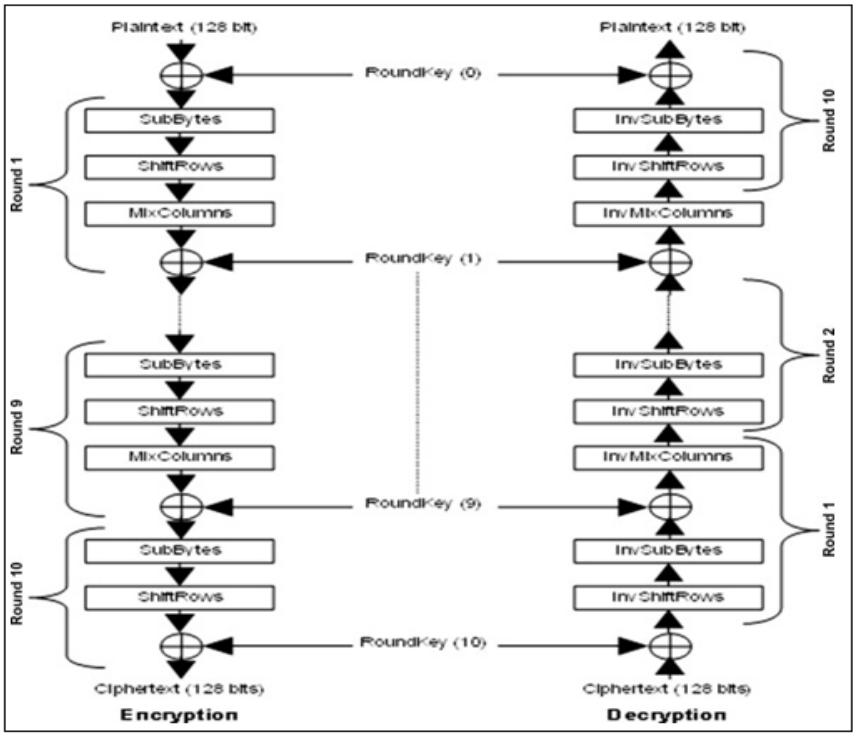

Fig 3. Rounds in AES

B. Key Distribution with Diffie-Hellman

DH (Diffie-Hellman) is a key distribution algorithm that helps two users to share secret key between them without the need to exchange the secret key. An overview of the algorithm is given below .

Key Distribution Algorithm

To share secret key the two devices A and B, agrees on public numeric constants p and g. Here p is any random prime number and g is the generator of p which less than p.

- Let x and y be the private keys of the devices A and B respectively, the selected x and y by A and B can any random prime number and must be less than p.
- Let $r1=g^x \bmod p$ and $r2=g^y \bmod p$ were computed by devices A and B respectively
- The computed r1 and r2 were exchanged between A and B.
- The end A computes $(r2)^x \bmod p$.
- The end B computes $(r1)^y \bmod p$.
- Since $K = g^{yx} \bmod p = g^{xy} \bmod p$, shared secret key = K.

IV. IMPLEMENTATION

First the input video is divided into a series of frames called P-Frame, B-Frame and I-Frame and only I-Frames are encrypted stream is transmitted along P-Frame and B-Frame to user. The whole system is implemented in Windows Azure Cloud platform. The frame set is shown in Fig4, Fig5 and Fig6.

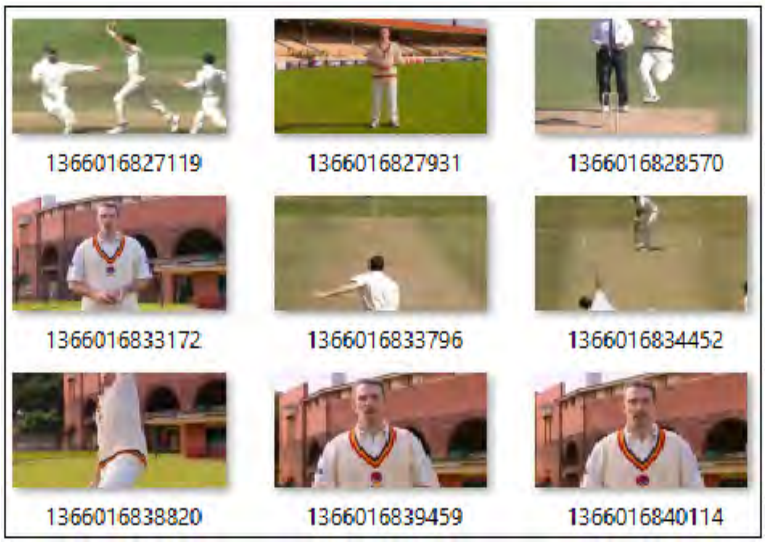

Fig 4. Frame Set

Fig 5. Frame Set

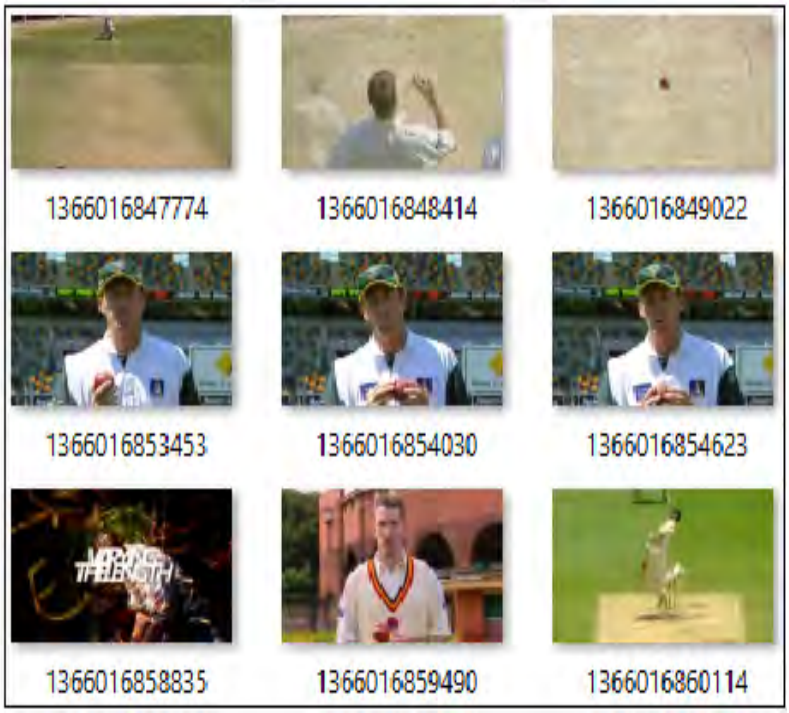

Fig 6. Frame Set

V. RESULTS

We have used MS Windows Azure Cloud Environment on Hp Proliant DL 385G Six Cor AMD Opeteron™ Proce with 32 GB of RAM and 500 GB HD in our experiment. For video transmission, we used UDP transmission protocol to send and receive the video packets through the network channel. Java programming language has been used since it has many advantages with the network programming. In addition, we modified the standard AES and RC4 codes to encrypt different lengths of video streams. We developed our final code in some functions that handle the encryption operations. We selected a fixed key length for AES, RC4 encryption algorithms. Fortunately, AES helps us to encrypt directly 128 bits of a video stream, which makes the computation fast in comparison to their work and finally, we examined the effect of encrypting the whole length of multimedia packets. Since the quality of service is very significant in multimedia networking, we have measured our system performance based on the time delay. where it is visible slightly in the transmission and reception of data. We will measure the delay in encrypting video packets for two algorithms. The result shows that AES gives better performance compared RC4 algorithm.

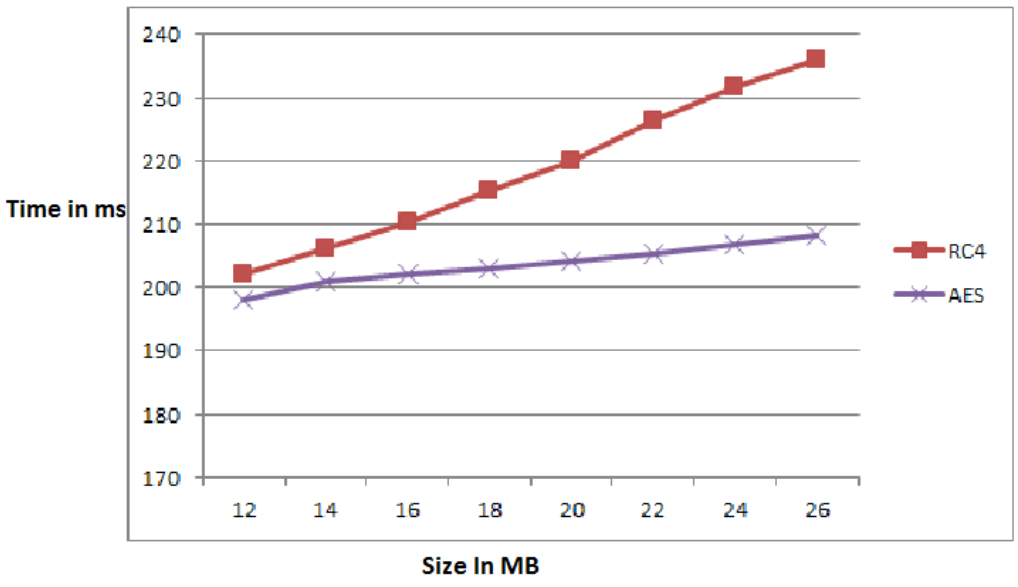

Fig 7. Comparison of RC4 and AES

VI. CONCLUSION

We used AES algorithm to encrypt the I-Frames of the video stream and it a faster compared to other algorithms because encryption and decryption is done on selective frames. It encrypts at most 128 bits, no matter what type of frame is used. Our proposed mechanism considerably reduces time for encryption and decryption.

REFERENCES

[1] Le Gall, Didier, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol.34, no.4, pp. 46-58, April 1991.

[2] NIST Home Page http://csrc.nistgov/encryption/aes.

[3] C. Shi, Sheng-Yih Wang, and Bharat Bhargava, "MPEG Video Encryption in Real-time using secret key cryptography", *Proc. Of PDPTA 99, Las Vegas, Nevada* 1999.

[4] C. Shi and Bhargava, "A Fast MPEG Video Encryption Algorithm", *Proceedings of ACM International Multimedia Conference, Bristol, UK*, pp. 81-88, September 1998.

[5] I. Agi and L. Gong, "An Empirical Study of MPEG Video Transmission", *Proceedings of the Internet Society Symposium on Network and Distributed Systems Security*, pp. 137- 144. San Diego, CA, Feb. 1996.

[6] Lei Tang, "Methods for Encrypting and Decrypting MPEG Video Data Efficiently", *Proceedingsof ACM Multimedia 96, pp. 219- 229, Boston, MA*, November 1996.

[7] Fuwen Liu, Hartmut Koenig. A survey of video encryption algorithms. Computer & Security 29(2010) ;3-15.