

Research on the Performance of Encryption Techniques for High-Bandwidth Multicast Video Streaming

Shanxi Li¹, Wenbo Chen^{2✉}, Zhihao Shang²

Information Technology Services of Lanzhou University
Lanzhou, P. R. China

¹ lisx@lzu.edu.cn

² chenwb@lzu.edu.cn

² shangzh11@lzu.edu.cn

Abstract-Recently, high-bandwidth streaming media security is becoming more important to the continuous increase of the wide application of multimedia and the improvement of image processing on the internet. However the tradition data encryption solutions are not appropriate for such a high-bandwidth streaming media. In spite of extensive research in multimedia security, the capability of symmetric encryption techniques in secure real time multicast video streaming has not been fully studied. In this paper, our aim is to study how to secure the streaming data transmission in the insecure network environment of multicast and broadcast. We consider the performance of AES, DES and RC4 encryption algorithms in encrypting the data stream with the bandwidth of 1~1000Mbps. MPEG-4 format of multicast streaming is used to test the effectiveness of the three encryption techniques. The simulation results showed that the AES and RC4 have the throughput over 250Mbps, can meet our demand currently.

Keywords: Video Streaming Security, Multicast, High-bandwidth, Encryption.

1 Introduction

Streaming media content is the fastest growing area of the internet. Broadcasters are offering live streams over the web, internet service providers are rolling out their own IP-based TV services. At the same time, Triple Play is the inevitable trend of the future computer network, high-speed internet access; television and telephone services over a single broadband connection. With a computer and a network cable can meet all your demands; watching TV, surfing on the internet, making telephone calls. Streaming media is getting more and more important. So it is very necessary to secure networked continuous media from potential threats such as hackers, eavesdroppers, etc. Nowadays the media streaming

technology has been widely applied, such as video conferencing, medical imaging systems, pay-per-view (PPV) and web-based channels (IPTV) [1]. Such systems use different types of encryption techniques to ensure the security of networked multimedia applications.

We choose three popular Symmetric Encryption Algorithms which are AES; DES and RC4. We want to know which encryption algorithm can cope with the video streaming data better? And what is the critical bandwidth of an encryption algorithm?

2 Video Encryption Algorithms

The video streaming system is shown in figure 1; it consists of several functional blocks. At the beginning, raw data will be read from the internet or local files then pre-compressed by the video compression algorithm. The encryption block will encrypt the compressed data and send it to the transport block. The data will be packet and spread to the internet. Packets may be dropped or experience excessive delay in the internet due to congestion. When packets delivered to the client successfully, first they should pass through the transport layer and then decrypted before decoded in a streaming media decoder [2].

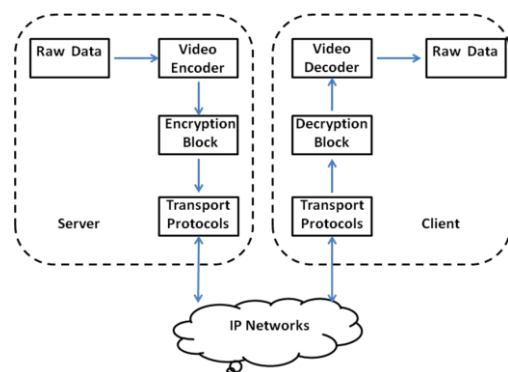


Figure 1: Video streaming system

The encryption of a video stream can be done in two ways [3]. The first technique is the secret key and the second is the public key encryption. Streaming media has a large amount of data and its transmission requires strong real time performance, but public key cryptography algorithms are too

complex to low efficiency, so public key cryptography is not applicable for secure real time video streaming.

The secret key cryptosystem has two branches: block cipher and stream cipher branch, both of them can be used to secure streaming media data. Block cipher is similar to the computational process both of them divided the data into blocks and then process every block. Stream cipher has faster encryption speed and simpler treatment. The secret key cryptography has many standard algorithms; each of them is different in their performance and security. When choosing a suitable algorithm, we should consider the application characteristics and the network condition and then tradeoffs between confidentiality and achievement.

2.1 Block Cipher Algorithms

Block cipher is a group of fixed lengths of the plaintext encryption algorithm. It packets the plaintext into a certain length of the block then encrypts the blocks of plaintext. After that, the key group arithmetic with the plaintext then we get the cipher text group. When decrypting the cipher text group and key group, we could restore the plaintext group. The basic principle is shown in figure 2.

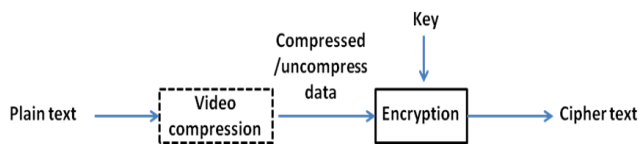


Figure 2: Basic principle of block cipher algorithms

The characteristic of the block cipher is that the key is fixed at a certain time instead of been transform all the time, so the key's dispensation will bring convenience. However, block cipher has a transmission error diffusion problem, so it cannot be used in poor channel quality casings. By the way, block cipher has two of the most famous algorithms they are DES (Data Encryption Standard) and AES (Advanced Encryption Standard) encryption algorithm.

The AES algorithm is essentially Rijndael^[5] symmetric key cryptosystem that processes 128-bit data blocks using cipher keys with a length of 128, 192, or 256 bits. Rijndael is more scalable, it can handle different key sizes and data block sizes, however they are not included in the standard. The basic blocks of AES operation are shown in figure 3. Further details about this algorithm can be found in^[6].

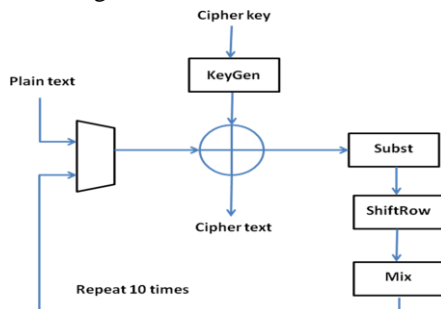


Figure 3: Basic blocks of AES operation

The DES algorithm exchanges the 64 bits of plaintext into 64 bits of cipher text, in which 8 bits are for parity and another 56 bits are cipher text. The DES 64 bits of the input data block will re-combination at first, and then the output data is divided into two parts L0 and R0, each part has the length of 32 bits and do some replacement before and after, finally we get the output which formed by the left 32 L0 and the right 32 R0. According to this rule, after 16 times of iteration operations we get the L16 and R16, and then import them to replace the initial permutation inverse, then the output is the cipher text. The basic blocks of DES operation are shown in figure 4.

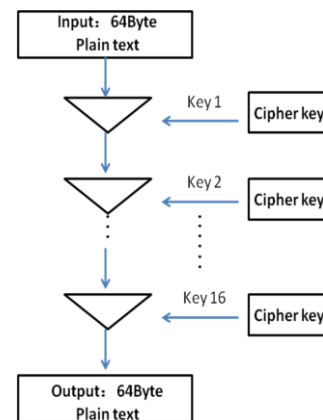


Figure 4: Basic blocks of DES operation

2.2 Stream Cipher Algorithms

If the block cipher algorithm fixed the length of the packet with a bit or a byte, and the stream cipher will encrypt the plaintext bit by bit or byte by byte. As for a stream cipher, the key is a plaintext of the same length with sequence. The encryption of the stream cipher is to arithmetic the plaintext sequence and the key stream sequences on bitwise, and the decryption key generated by the synchronized flow reverse transformation^[4]. The basic principle is shown in figure 5.

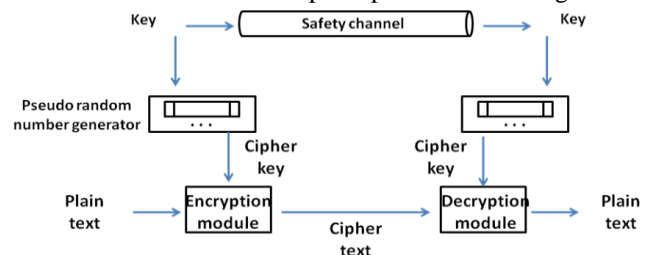


Figure 5: Basic principle of stream cipher algorithms

Comparing with the ordinary block cipher and public key cryptography, the stream cipher is faster, and has the best real time property. Therefore, Stream cipher is suitable for a large amount data and high requirement of real time streaming media encryption. So it is also the military and the diplomatic field application of a mainstream cryptographic system.

RC4 is Stream cipher structure in which it encrypts plain text one byte at a time with variable length key size from 1 to 256 bytes (8 to 2048). The principle of RC4 algorithm is "disturbed", it includes initialization algorithm and pseudo-random cipher algorithm. During the initialization process, the main function of the key is to randomly scramble a 256 byte of the initial number of clusters; a different number of clusters after a pseudo-random cipher algorithm's processing we can get the sub key sequence. At the end, the key sequence XOR (Exclusive OR) with the plaintext, then we get the cipher text^[7]. The basic operation and sequence of RC4 is shown in figure 6.

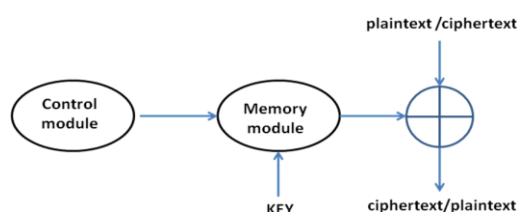


Figure 6: Basic blocks of RC4 operation

Encrypted using the RC4 algorithm is the XOR way, when the sub-key sequence repeated; the cipher text is possible to be cracked. However there is no possibility that the length of 128 bits RC4 key to be duplicated. So the RC4 is still one of the most secure encryption algorithms.

3 Previous Work in Video Encryption

There have been many researchers attempt to secure media stream. The simplest way is to encrypt the entire stream data using standard encryption algorithms. In fact, researchers have used the naive algorithm to approach^[6]. The greatest concern about this approach is the speed of processing due to the large size of stream data. Another method to secure MPEG streams is the selected encryption algorithm which encrypts only the I-frame of MPEG streams^[8,9]. Meyer and Gadegast^[10] have designed a new MPEG-like bit-stream SECMPG that incorporates selective encryption and additional header information, and has high-speed software execution. SECMPG can use both DES and RSA and implements four levels of security: first level encrypts all headers. Second level encrypts all headers plus the DC and lower AC terms of the I-blocks. Third level encrypts I frames and all I-blocks in P and B frames. Forth level encrypts all data. SECMPG is not compatible with standard MPEG. A special encoder/decoder would be required to view unencrypted SECMPG streams. A proposal targeting at the integration of compression and encryption of MPEG streams into one step is presented in^[13] using the "Zigzag Permutation Algorithm", where the basic idea is to use a random permutation list to replace the zigzag order to map the individual 8*8 block to a 1*64 vector.

Sale^[12] studied the performance of encryption and decryption algorithms such as AES for real time video streams. He adapted AES and XOR algorithms to be used with JPEG, H261, CellB, and MPEG-1/2 video encoders and

decoders. He attempted to select specific frames to encrypt. The encrypted video streams are combinations of I, P, and B frames. In^[13], four fast MPEG video encryption algorithms are presented. These algorithms are based on the DES [3] by using a secret key to randomly change the sign bits of Discrete Cosine Transform (DCT) coefficients and/or the sign bits of motion vectors. The encryption is accomplished by the inverse DCT (IDCT) during the MPEG video compression processing. These algorithms add a small overhead to the MPEG CODEC. As can be noticed the previous authors haven't studied the performance of the AES in encrypting MPEG-4 video streaming. Moreover, most studies haven't used peer to peer platforms to transfer the video stream which has gained more interest in recent decayed due its wide application spectrum.

4 Performance and Analysis

How to measure the performance of a good encryption algorithm, the ideal streaming media encryption algorithm should have the following features^[8]:

- Encryption process costs should be as small as possible and the speed should be as quickly as possible;
- Encryption algorithm should not reduce the media's original quality and compression ratio;
- Encryption algorithm should be safe enough and can withstand common multimedia analysis and recovery technology.

The whole process of the raw data has been sent by the server until reaching the client and has been reduced, just as shown in Fig.1, the time delay included: encryption time (Te), transmission time (Tt) and decryption time (Td).

We can assume that the time delay T represents the summation of the previous time delays ($T = T_e + T_t + T_d$).

Finally, we take the three encryption algorithms' critical bandwidth into consideration.

5 Results

We have used Linux machines with Intel® E5405 CPU 2.0 GHz, 4GB of RAM in our experiments. For video transmission, we used UDP transmission protocol to send and receive the multicast video packets through the network channel. Python programming language has been used since it has many advantages of the network programming. In addition, we modified the standard AES, DES and RC4 codes encrypt different lengths of video streams. We develop our final code in some functions to handle the encryption operations. We selected a fixed key length of 128 bits for AES, RC4 and 56 bits for the DES encryption algorithm. We have measured our system performance based on the delay where it is visible slightly in the transmission and reception of data.

Firstly, we stimulate the experiments of the article [14], tested the performance of the three encryption algorithms in our experimental environment. We have measured the time delay in an encrypting number of texts, audio and video packets with the three algorithms mentioned before.

For text, we have measured T_e to encrypt 10000 packets, 15 bytes each. Figure 7 shows the time for different encryption algorithms (AES, DES and RC4) for the MPEG-4 text.

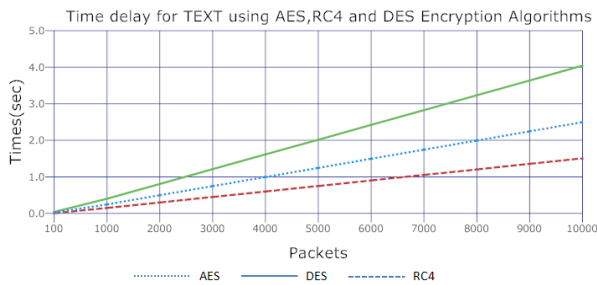


Figure 7: Time delay T_e for TEXT using AES, DES and RC4 Encryption Algorithms

For audio, we have measured T_e to encrypt 200 packets, 500 bytes each. Figure 8 shows the time for different encryption algorithms (AES, RC4 and DES) for the MPEG-4 audio.

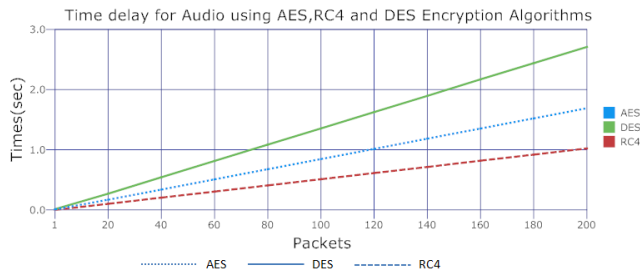


Figure 8: Time delay T_e for AUDIO using AES, DES and RC4 Encryption Algorithms

For video, we have measured T_e to encrypt 100 packets, 1024 byte each. Figure 9 shows the time for different encryption algorithms (AES, DES and RC4) for the MPEG-4 video.

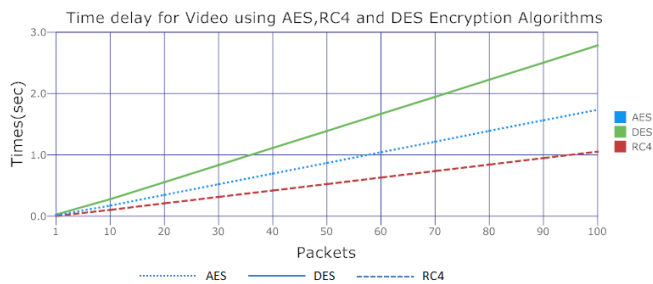


Figure 9: Time delay T_e for VIDEO using AES, DES and RC4 Encryption Algorithms

As shown in Figure 7, 8, 9 the overhead time of encrypted packets using RC4 is less than the overhead time using AES and DES. And the text time delay is longer than the audio and video, but the audio and video time delay is almost the same. In the three experiments the packet transparent speed has been fixed. Text encryption has low ratio transmission, the time mainly cost of waiting for the packets. The performance of audio and audio encryption can't

meet the network transport speed, the time mainly cost at the encryption.

Secondly, we have measured the $(T_e + T_t)$ of the three encryption algorithms in the condition of the bandwidth from 1Mbps to 1000Mbps. Receiving and encrypting 10000 packets of 1324 bytes each use the protocol of UDP. Figure 10 shows the time delay for AES, DES, RC4 and Blank (the time delay just receiving without encrypting) encryption algorithms.

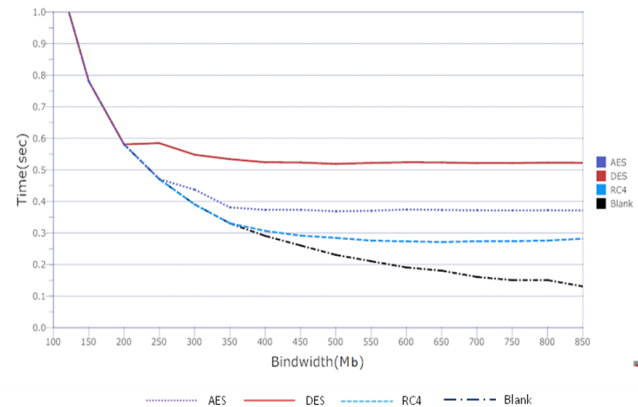


Figure 10: Time delay T_e for multicast using AES, DES, RC4 and Blank encryption algorithms

From the Figure 10 we can tell the top bandwidth of the three algorithms, DES gets its critical bandwidth of 200Mbps; AES gets its critical bandwidth of 250Mbps and RC4 gets its critical bandwidth of 350Mbps. The Blank shows the time cost in transport 10000 packets of the different bandwidth. The three encryption algorithms can't stand more than their critical bandwidth, after which the three encryption algorithms at their top usage.

Finally, a comparison between the selected encryption algorithms is conducted from the view of safe time. The result of this comparison is shown in Figure 11. This figure indicates the great difference between AES and other algorithms. This implies that AES can be considered the best one from the point of safe time ^[14].

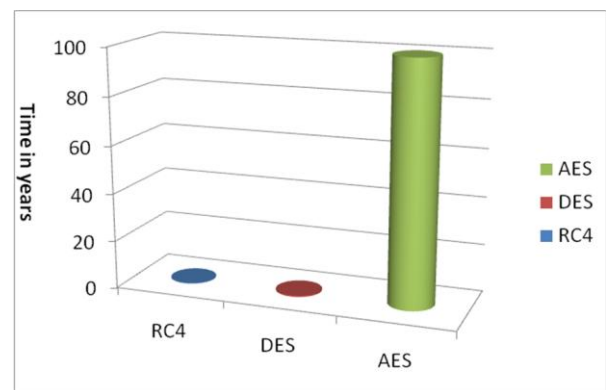


Figure 11: Encryption techniques safe time for AES, DES and RC4 encryption algorithms

6 Conclusion

Our study showed that the AES and RC4 encryption algorithms can be used to encrypt high-bandwidth of 20Mbps streaming media effectively. The encryption delay overhead using RC4 is less than the overhead using AES and DES algorithms, but AES is much safer than RC4. Therefore, we conclude that both of AES and RC4 can secure high-bandwidth real time streaming, AES gets much more safety and RC4 get much more bandwidth.

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] Chun-Shien L, "Multimedia Security Steganography and Digital Watermarking Techniques for Protection of Intellectual Property", Idea Group Publishing 2005.
- [3] William Stallings, "Cryptography and Network Security, Principles and Practice", Pearson education, Third Edition, 2005.
- [4] Ueda N. Optional Linerar Coom bination of Neural Networks for Improving Classication Performance [C]. IEEE Trans. Pattern Anal. Machine Intell. , 22(2): 127-130.
- [5] B. Gladman, "A Specification for Rijndael, the AES Algorithm," (<http://fp.gladman.plus.coml>, 2001).
- [6] I. Aging and L. Gong, "An 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.
- [7] T. Seidel, D. Sock, and M. Sramka, "Cryptanalysis of Video Encryption Algorithms", Proceedings of the 3rd Central European Conference on Cryptology TATRACRYPT 2003.
- [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] T. B. Maples and G.A. Spanos, "Performance Study of a Selective Encryption Scheme for the Security of Networked Real-time Video", In Proceedings of lath International Conference on Computer Communications and Networks, Las Vegas, Nevada, September 1995.
- [10] J. Meyer and F. Gadegast, "Security Mechanisms for Multimedia Data with the Example mpeg-1 Video", Available at WWW via <http://www.powerweb.de/phad-e/phade.html>. 1995.
- [11] L. Tang, "Methods for Encrypting and Decrypting MPEG Video Data Efficiently", In Proceedings of The Fourth ACM International Multimedia Conference (ACM Multimedia'96), pages 219-230, Boston, MA, November 1996.
- [12] Salah Aly, Multimedia Security: Survey and Analysis, Multimedia and Networking Research Lab, CTI, DePaul University, Chicago, 2003.5, available: www.depauledu/seminaffspr2003/secvideo.pdf.
- [13] B. Shi, W. Changgui and S. Wang, "MPEG Video Encryption Algorithms", Multimedia Tools and Applications, Vol.24, Issue: I, pp. 57-79, September 2004.
- [14] Wail S. Elkilani, Hatem M. Abdul-Kader. Performance of Encryption Techniques for Real Time Video Streaming. IEEE, 978-1-4244-3778-8/09.