Performance of Encryption Techniques for Real Time Video Streaming

W.S. Elkilani and H.M. Abdul-Kader
Faculty of computers and Information- Menufiya University- Shabin ElKom-Egypt
welkilani@gawab.com , hatem6803@yahoo.com

Abstract *Recently, multimedia security is becoming more important with the continuous increase of digital communications on the internet. Moreover, special and reliable security is needed in many digital applications (such as video conferencing and medical imaging systems). The classical techniques for data security are not appropriate for the current multimedia usage. As a result, we need to develop new security protocols or adapt the available security protocols to be applicable for securing the multimedia applications. Encryption of MPEG-4 video streaming using AES has not been studied. In this paper, the performance of AES in encrypting MPEG-4 video is considered. The performance of AES is compared to two symmetric encryption techniques namely; RC4 and XOR. Three data types (text, audio and video) are used to test the effectiveness of AES in encrypting MPEG-4 . Simulations showed the efficiency of the AES encryption technique in such application for the different data type given.*

Keywords*: Multimedia Security, Video Streaming, AES, MPEG-4*

1. Introduction

The advent of networked multimedia system systems will make continuous media stream. It is very important to secure networked continuous media from potential threats such as hackers , eavesdroppers , etc . The applications for streaming are endless. Streaming can be delivered as a complete video package of linear programming, as a subscription service, or as pay-per-view (PPV). It can form part of an interactive web site or it can be a tool, in its own right, 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 [1][2].

Playing video streams over a network in a real time requires that the transmitted frames are sent in a limited delay. Also, video frames need to be displayed at a certain rate; therefore, sending and receiving encrypted packets must be sent in a certain amount of time in order to utilize the admissible delay. For example: Video on-Demand requires that the video stream needs to be played whenever the receiver asks for it. So, there are no buffer or playback concepts for the video stream (i.e. it runs in real time). Hence, there are many challenges for multimedia security such as:

- The natural size of multimedia data after compression is usually very large, even if using the best available compression techniques. The size of a two-hour MPEG-1 video is about 1 GB.
- Future applications of multimedia need to be run in real time on processes such as video on demand.
- Performance of processing multimedia streams should be acceptable (i.e. bounded by certain value of delay).
- The encryption techniques should be fast enough and require a small overhead in comparison to compression techniques.

The goal of this research is to focus on the following points. First, implementing AES for MPEG-4 in a real time secure video transmitting system. Second, comparing the performance of the AES with respect to two major encryption techniques over a peer to peer channel. Third, evaluating the difference between the overhead resulting from different data types in multimedia (text, audio, and video) due to the three encryption techniques (Xor, RC4, AES).

This paper is organized as follows. In section 2, the basic concepts of video encryption techniques are given. Then in section 3 a brief overview of the previous video encryption methods is discussed. Video streaming quality of services is shown in section 4. In section 5 results of using different encryption techniques are shown. Finally the conclusion is drawn in section 6.

2. Basic Concepts of Video Encryption

The encryption and decryption of a plain text or a video stream can be done in two ways. The first technique is the secret key encryption. The second technique is the public key encryption [3][4]. Public key cryptography is not applicable for secure real time video conferencing because its operations require an amount of time, which is not suitable for video conferencing.

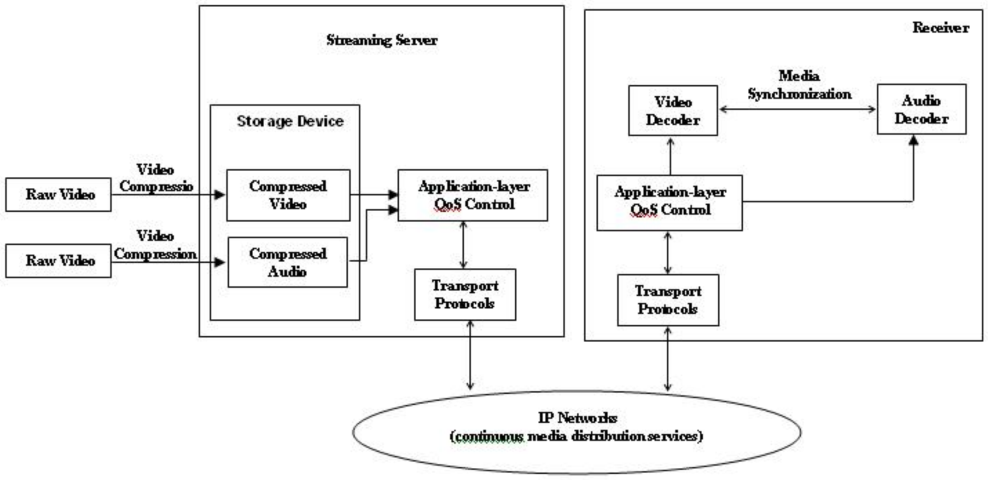

Figure 1. One way data flow block diagram for captured multimedia devices

A video streaming system typically consists of seven building blocks, as illustrated in Figure1. In this figure, raw video and audio data are pre-compressed by video compression and audio compression algorithms and then saved in storage devices. Upon the client's request, a streaming server retrieves compressed video/audio data from storage devices and then the application-layer QoS control module adapts the video/audio bit-streams according to the network status and QoS requirements. After the adaptation, the transport protocols packetize the compressed bit-streams and send the video/audio packets to the Internet IP networks. Packets may be dropped or experience excessive delay inside the Internet due to congestion; on wireless IP segments, packets may be damaged by bit errors. To improve the quality of video/audio transmission, continuous media distribution services are deployed in the Internet. For packets that are successfully delivered to the receiver, they are first pass through the transport layers and then are processed by the application layer before being decoded at the video/audio decoder. To achieve synchronization between video and audio presentations, media synchronization mechanisms are required [5].

There are many video encryption algorithms. Such encryption techniques can be classified as follows: naive algorithm, selective algorithm, Zig-Zag algorithm, RC4 and AES [6]. A review for each one is briefly given. The idea of naive encryption [3] is to deal with the video streams as text data. The simplest way to encrypt video streams is to encrypt every byte. Naive algorithm encrypts every byte in the whole video stream. Naive algorithms guarantee the most security level. However, it is not an applicable solution if the size of the data is large. Due to encryption operations, the time delay increases and the overhead will not be satisfactory for the real time video encryptions.

In selective algorithm [4], four levels of selective algorithms are suggested. These four levels are encrypting all headers, encrypting all headers and I (initial) frames, encrypting all I frames and all I blocks in P and B frames, and finally encrypting all frames as in Naïve algorithm to guarantee the highest security. The idea of ZIG-ZAG algorithm [4] is basically encrypting the video streams before compressing them. Explicitly, when mapping the 8x8 block to a 1x64 vector each time in the same order. We can use a random permutation to map this transformation of the 8x8 block to the 1x64 vector. Therefore, the concept of the encryption key does not exist in the ZIG-ZAG permutation algorithms. Once the permutation list is known, the algorithm will not be secure any longer.

A new video encryption algorithm called VEA that depends on dividing the video streams into chunks. These chunks are separated into two different lists (odd and even lists). Afterward, applying an encryption algorithm like DES to the even list and the final cipher is concatenation of output of encryption algorithm XOR with the odd list streams [5-6]. 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) . It is a symmetric encryption algorithm in which the same key is used for encryption and decryption .The algorithm is based on the use of random permutation. RC4 is the most widely used stream cipher. It is used in the SSL/TLS (Secure Socket Layer/Transport Layer Security) standards that have been defined for communication between web browsers and servers. It consists basically of two main operations, namely key Setup operation and ciphering operation. In the first operation RC4 generates a pseudorandom stream of bits (a "keystream") then applying some kind of operation on key such as permutation and expansion so as to be more randomized [3]. While in the second operation the plaintext is Xored with the key. The basic operation and sequence of RC4 is shown in figure 2. Further details about this algorithm can be found in [6].

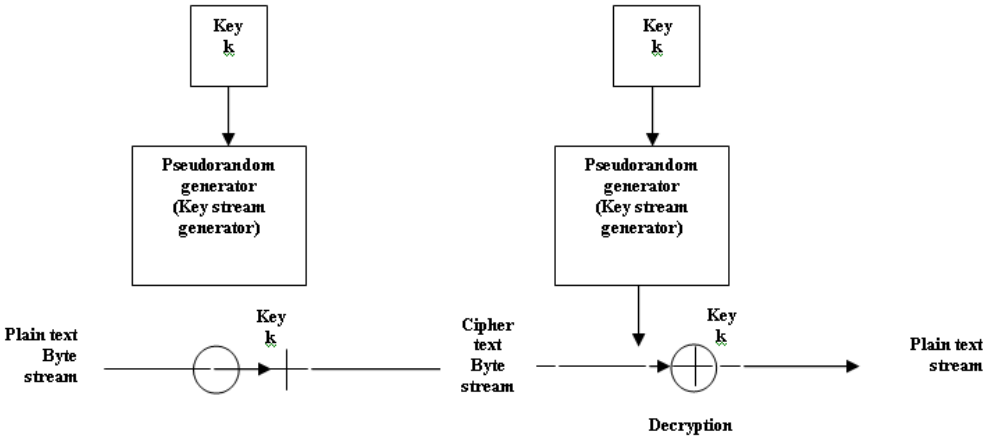
Key
k
Pseudorandom generator (Key stream generator)
Plain text Byte stream
Key
k
Key
k
Pseudorandom generator (Key stream generator)
Cipher text Byte stream
Key
k
Plain text stream
Decryption

Figure 2. Basic architecture of RC4

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].

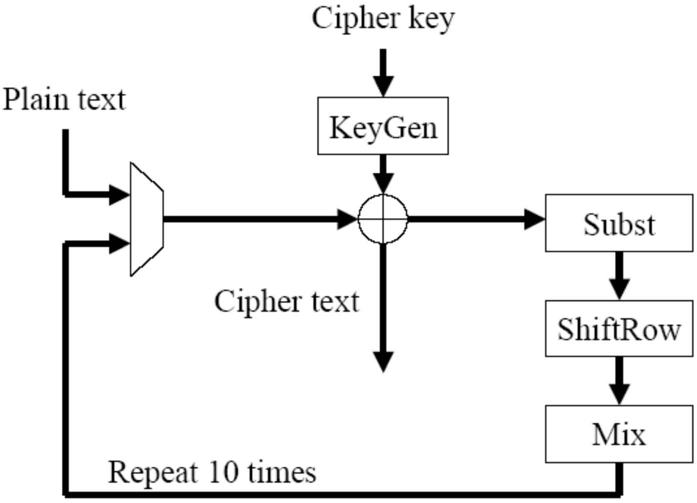

Figure 3. Basic architecture of AES

3. Previous Work in Video Encryption

Some proposed attempts to secure MPEG streams have been reported. The most straightforward method is to encrypt the entire MPEG stream using standard encryption methods. In fact, they have used the naive algorithm approach [8]. The greatest concern about this approach is the speed of processing due to the large size of MPEG files. Another method to secure MPEG streams is the selective encryption algorithm which encrypts only the I-frame of MPEG streams [9,10]. Meyer and Gadegast [11] have designed a new MPEG-like bit-stream SECMPEG that incorporates selective encryption and additional header information, and has high-speed software execution. SECMPEG can use both DES and RSA and implements four levels of security: 1st level— encrypts all headers. 2nd level — encrypts all headers plus the DC and lower AC terms of the I-blocks. 3rd level — encrypts I frames and all I-blocks in P and B frames. 4th level — encrypts all data. SECMPEG is not compatible with standard MPEG. A special encoder/decoder would be required to view unencrypted SECMPEG streams. A proposal targeting at integration of compression and encryption of MPEG streams into one step is presented in [12] using the "ZigZag-Permutation Algorithm", where the basic idea is to use a random permutation list to replace the zig-zag order to map the individual 8x8 block to a 1x64 vector.

- Salah [13] studied 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 [14], 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 decompression processing. These algorithms add a small overhead to the MPEG codec. As can be noticed that 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 video stream which has gained more interest in recent decayed due its wide application spectrum.

4. Video Streaming Quality of Service

In this section, we will show the parameters used to measure the quality of an encryption technique. QoS refers to the ability of a network to provide better service to selected network traffic over various underlying technologies. The main QoS features that provide better and more predictable network service can be summarized in the following:

- Algorithm setup time (Ts): Similar to key setup time, the algorithm setup time reports the minimum amount of time before an algorithm is ready to process data. Time to create look-up tables, etc. will fall in this category. None of the evaluated algorithms contained an algorithm setup time greater than zero.
- Time to encrypt one block (Te): This parameter will address minimum latency times for each of the algorithm submissions. The time to encrypt one block , measured in nanoseconds, is a function of two parameters: the worst-case path delay between any two registers, and the number of rounds in the algorithm.
- Time to decrypt one block (Td): As above, this parameter will address minimum latency times for each of the algorithm submissions. Decryption does not always require identical processing as encryption. Therefore, the time required to decrypt one block is reported.
- Time to switch keys (Ts): Originally, this parameter was included as a measure to encompass both key setup time and algorithm setup time overhead. However, since none of the evaluated algorithms contained an algorithm setup time, this parameter is identical to key setup time. Therefore, it will not be reported further in this document.

We can assume that the time delay T represents the summation of the previous time delays (T=Ts+ Te + Td +Ts)

5. Results

We have used the Windows machines with Intel® Celeron CPU 3 GHz, 2 MB of RAM in our experiments .In addition, we used 300SC-Y web camera to capture the video frames. For the video transmission, we used UDP transmission protocol to send and receive the video packets through the network channel. Visual C++ 6.0 programming language has been used since it has many advantages with the network programming. In addition, we modified the standard AES and XOR 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 and XOR 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 QoS is very important in multimedia networking, we have measured our system performance based on the delay where it is visible slightly in the transmission and reception of data. We will measure the delay in encrypting number of text, audio and video MPEG-4 packets for the three algorithms. The calculations are based on the difference between the start of transmission and the reception time for 100000 packets (15 byte each packet). Figure 4 shows the time for different encryption algorithms (XOR, RC4 and AES) for the MPEG-4 text data type.

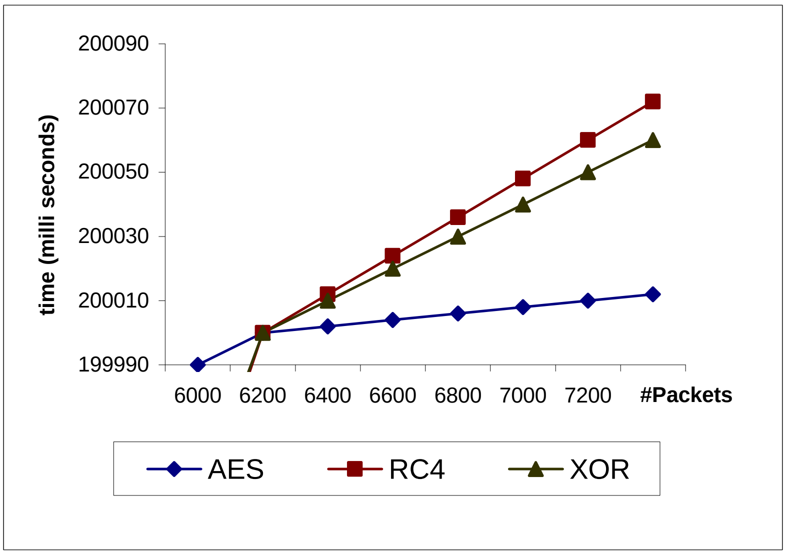

Figure 4. Time delay T for TEXT using XOR, RC4 and AES Encryption Algorithms

For audio, we have measured Te to encrypt 500 packets, 600 byte each. Figure 5 shows the time for different encryption algorithms (XOR, RC4 and AES) for the MPEG-4 audio.

.

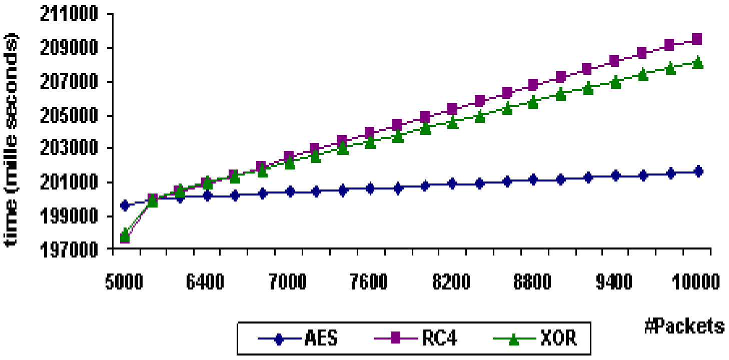

Figure 5. Time delay T for AUDIO using XOR, RC4 and AES Encryption Algorithms

For video, we have measured the time (Te) to encrypt 500 packets, 2464 byte each. Figure 6 shows the time for different encryption algorithms (XOR, RC4 and AES) for the MPEG-4 video

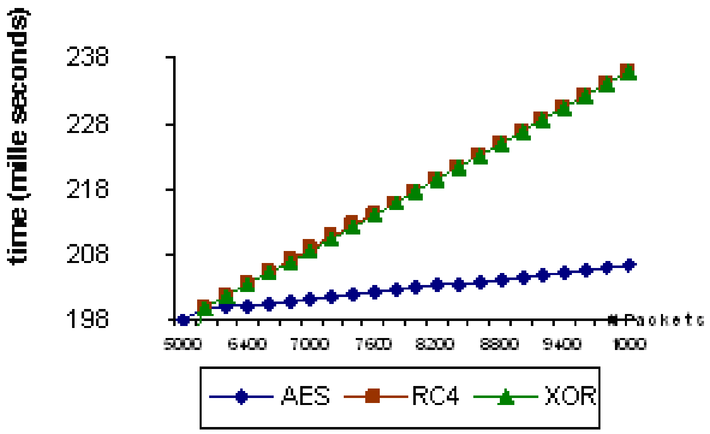

Figure 6. Time delay T for VIDEO XOR, RC4 and AES Encryption Algorithms

As shown in figures 4, 5 and 6, the overhead time of encrypted packets using AES is less than the overhead time of the encrypted packet using RC4 and XOR. Also the overhead time of encrypted packets of type text is less than the overhead time of the encrypted packet of type audio then video. So we use 100000 packet of type text each of size data 15 byte to be sensible of low rate transmission, then audio 500 packet of size 600 byte, then video 500 packet of size 2464 byte and very high rate to avoid video flickering which cause very high overload. From these Figures, the relative time spent for the encryption operation using AES does not negatively affect video stream transmission. Second this is acceptable for video transmissions. In conclusion, encrypting MPEG-4 video streams using AES is an applicable solution to secure real time video transmission. Based on the above results and the criteria which differentiate between different encryption techniques we can summarize the result as shown in table 1

Algorithm	Cost	I/O	T to Encrypt B	T to Decrypt B
XOR	Low	Fixed	Stream Cipher	Stream Cipher
RC4	Medium	Fixed	Stream Cipher	Stream Cipher
AES	Very High	Fixed	0.01ms (depend on the block size)	0.01ms (depend on the block size)

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 7. This figure indicates the great difference between AES and other algorithms. This implies that AES can be consider the best one from the point of safe time.

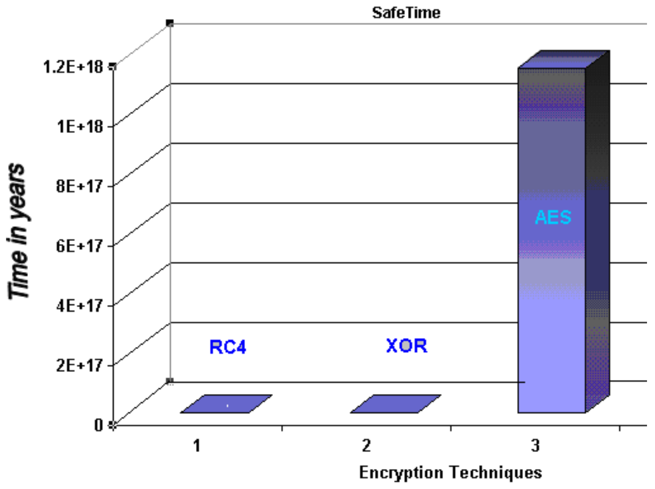

Figure 7. Encryption techniques Safe time for

6. Conclusions

Our study showed that the AES encryption algorithm can be used effectively to encrypt MPEG-4. The performance of AES encryption frames is sufficient to display the received frames on time. The encryptions delay overhead using AES is less than the overhead using RC4 and XOR algorithm. In addition, AES can achieve satisfactory encryption results with little overhead. Therefore, we conclude that using AES in encrypting MPEG-4 is a feasible solution to secure real time video transmissions.

7. 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”, IASTED 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] B. Gladman, “A Specification for Rijndael, the AES Algorithm,” (http://fp.gladman.plus.com/, 2001).

[8] I. Agi 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.

[9] 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.

[10] 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.

[11] J. Meyer and F. Gadegast, “Security Mechanisms for Multimedia Data with the Example mpeg-1 Video”, Available on WWW via http://www.powerweb.de/ phade/phade.htm/, 1995.

[12] 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.

[13] Salah Aly, “A Light-Weight Encrypting For Real Time Video Transmission”, TR04-002, College of computing and digital media, Depaul university, August 2004 (http://facweb.cs.depaul.edu/research/TechReports/ TR04-002.pdf)

[14] B. Shi, W. Changgui and S. Wang, “MPEG Video Encryption Algorithms”, Multimedia Tools and Applications, Vol. 24, Issue: 1, pp. 57-79, September 2004.