25891

Lightweight and Cost-Effective MPEG Video Encryption[1]

Lee Sian Choon, Azman Samsudin, and Rahmat Budiarto
Computer Science School, Universiti Sains Malaysia
11800, Penang, Malaysia
sianchoon@hotmail.com, azman@cs.usm.my, rahmat@cs.usm.my

Abstract

MPEG video structure and compression and various applications of encryption methods for this standard format are examined. Appreciation of MPEG is crucial to implement a well-suited encryption method for MPEG stream. Appropriate considerations in term of assignment scheme, applied block size and selective implementation can be taken into account in order to tune up the encryption performance. Three different approaches, which are permutation, XOR template and hybrid of both methods, are proposed. The performance of the suggested methods is verified in term of speed, picture invisibility, encrypted stream size (effect of compression ratio), and vulnerability to attacks (security). The purposed method is designed to work at the macroblock level. Proposed encryption method is believed to be secured and fast. All the experiments are applied using MATLAB programming due to its simplicity in data manipulation.

Keywords: *MPEG Video Encryption, Video Content Requirement, Security Enhanced MPEG, and Video Encryption Algorithm.*

1. Introduction

The emerging of multimedia and mobile communication technology has upshot the flow of media stream such as real-time video and audio contents across the Internet. To avoid the exposure of the stream data especially confidential information, protection is required to keep it away from the intruders. Standard encryption methods that currently exist could not be applied efficiently because those encryption methods are not suitable for large size data and typically for on-demand real-time processing. Moreover, low powered handheld devices lacking of buffer storage as well as computation power to store, decrypt and decode the encrypted video stream in ordinary will not be able to take advantage of secure media stream.

In addition, the information rate of multimedia content is high, but its value could be low. The content might have the situation where it is priceless for a short while, but useless afterward. It is worthless to spend costly computational power complex encryption method that was designed to protect the data for a long period of time. Thus, lightweight and cost-effective encryption method is targeted as the ideal method. In this paper three different approaches, which are permutation, XOR template and hybrid of both methods, are proposed. The purposed method is designed to work at the macroblock level to achieve the desired performance.

2. Related Work

Several compression methods have been identified by several working group and successfully increased the efficiency of video stream decoding process. One of the groups, MPEG working group, has invented MPEG standard that provides one of the popular and effective compression methods that greatly fitted to video stream. In this section, we are going to revisit MPEG protocol.

Moving Picture Experts Group (MPEG) was established in 1988 [1]. This group was formally known as ISO/IEC JTC1/SC29/WG11. Generally, MPEG standard is classified into three major parts: MPEG Video Coding, MPEG Audio Coding and MPEG Systems for synchronization of video and audio signals. As for this paper, we will be focusing on video stream of the MPEG Video Coding part. The core part of MPEG Video Coding is further divided into MPEG video structure and MPEG compression.

2.1 MPEG Video Structure

Generally, MPEG standard use a hierarchical composition to define the structure of MPEG video stream. Specified unique header or/and trailer are used to acknowledge the beginning and ending of particular levels accordingly. Each layer has its specification-defined information with specified length. All the layers that are used for describing the structure of MPEG video from the lowest, block layer to the highest, video

[1] The authors acknowledge the research grant provided by Universiti Sains Malaysia that has resulted in this article.

sequence layer can be referred to [2]. Figure 1 illustrates the overview structure of MPEG video stream as described above.

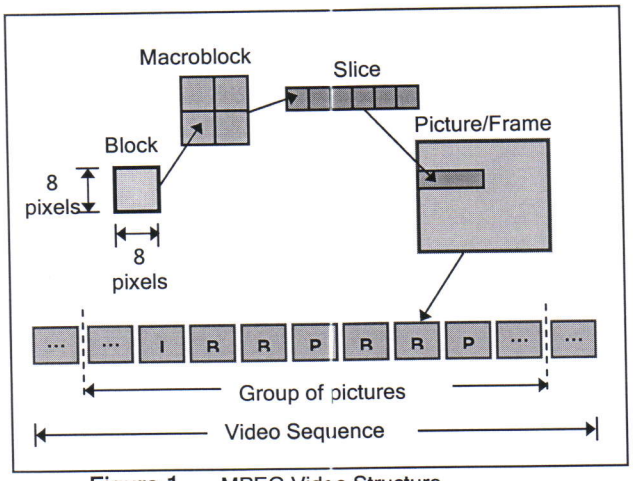

Figure 1. MPEG Video Structure

2.2 MPEG Compression

Transform coding is the most basic compression technique that applied to MPEG video stream. It is commonly known as intra coding because all the I-pictures are being compressed by this technique in order to reduce the spatial redundancy in the picture without referencing to other pictures. Figure 2 illustrates the flow chart of the transform coding. MPEG not only makes use of the spatial domain but also the frequency domain simultaneously to produce a powerful compression method. The details of other MPEG compression methods can be referred to [2] and other MPEG related resources.

Apparently, MPEG gives freedom to the encoder manufacturer in choosing parameters such as the frequency of the video sequence, the number of presence for each I-picture, P-pictures, B-pictures, quantization matrix etc.

2.3 Security Enhanced MPEG Player

Security enhanced MPEG Player (SE_MPEG) [3] is a secure video encoder and player based on Berkeley's MPEG Player and Zimmerman's PGP where the encryption of MPEG video files is processed frame by frame with extended PGP to deal with buffered frames. It is software-only implementation. This player intended to provide protection for multimedia applications such as personal video streams, videoconferences, and collaborative sensitive video data. Various choices of encryption may be performed in a protection hierarchy on only I-pictures, or both I-pictures and P-pictures, or all I-pictures, P-pictures and B-pictures. The overheads incurred are increased depending on the MPEG frame size, encoding format, and encryption method used.

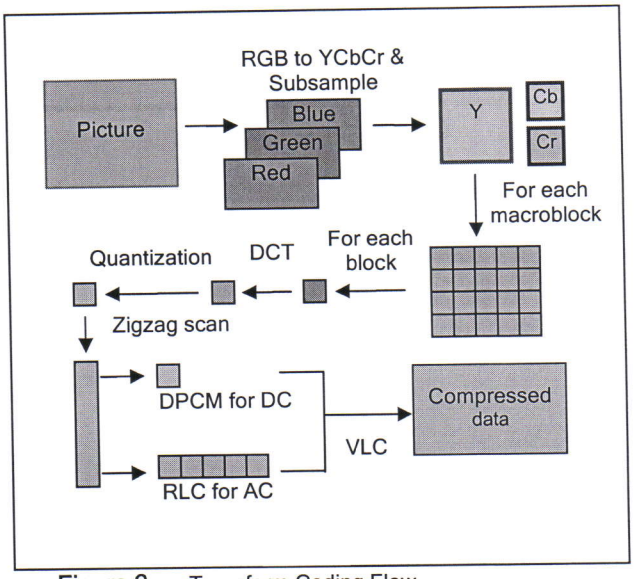

Figure 2. Transform Coding Flow

The analysis denoted that:

i. The increased protection is traded off against more overheads as more encryption is done.
ii. Better compression means less data as well as longer dependencies between MPEG pictures. Thus, less cryptographical overhead leads to increased compression.
iii. The display speed is increased when P-pictures or B-pictures are increased (corresponding to a better compressed file).
iv. Encrypting an I-picture has more effect in MPEG groups that have more P-pictures and B-pictures.
v. Encrypting video frames on the fly (without encryption beforehand) is recommended to avoid client waiting while the whole video file is being encrypted and the problem of transmission errors.
vi. Trade off between playback latency and CPU overhead for each picture processing against one another has to be considered. If multiple encrypted video streams is transmitted using on-the-fly encryption, the CPU overhead is extremely high.

As the result, SE_MPEG player fulfill the real-time playback requirement for Internet applications in term of frame rate, video quality and overall performance. This research work is essentially a hybrid of an existing popular video player with an existing established secure communication system. It has proved that the existing security method could be applied with some modifications in only software implementation. However, there are some other related works that are more specifically conducted to enhance the MPEG performance.

2.4 Existing MPEG Encryption

Several encryption algorithms that have been implemented to secure MPEG video in the past include Naïve algorithm, Selective algorithm (in different variations), Zigzag permutation algorithm, Pure permutation algorithm, and Video Encryption algorithm (VEA) [4,5,6,7,8,9]. Table 1 briefly summarizes them in assessable form.

Table 1. Comparison of MPEG Encryption Methods

Criteria / Method	Security level	Speed	Encrypted Stream Size
Naive	High	Slow	Remain
Only I-picture encryption	Low	Fast	Increase
I-picture & I-blocks in P- & B-picture encryption	Moderate	Fast	Increase
Sign-bit encryption	Moderate	Very fast	Remain
Improved I-MB & header of P-MB encryption	High	Fast	Increase
Zigzag permutation	Very low	Very fast	Significant increase
Pure permutation	Low	Very fast	Remain
VEA	High	Fast	Slightly increase

Most of the recent encryptions are compatible with MPEG whereby the particular encryption will not destroy the original MPEG architecture. The encrypted MPEG stream is able to recognize by MPEG standard decoder or player. From the table, we observed that:

i. Permutation method is the most efficient in term of speed but less secure.
ii. Selective algorithms have been greatly analyzed and used in MPEG encryption because of the characteristic of the MPEG standard.
iii. Instead of using MPEG picture structure layer, macroblock level is another layer structure that believed can enhance the MPEG encryption performance.
iv. Most of the encryption methods are designed to be compatible with MPEG standard even the streams are encrypted.

3 Proposed Design Methodology

From the literature review, we are motivated by the special requirement to design better encryption method particularly for video content. Subsequently, we are introduced to MPEG standard, which includes its extraordinary layered structure and effective compression method in depth. We then study the strength and weakness of several MPEG encryption methods in its variations.

Following are two different approaches suggested based on the state of art of encryption, the characteristic of MPEG and the existing MPEG encryption methods. Both approaches are premeditated to work at macroblock layer.

3.1 Selective XOR Template

The first approach is performing XOR operation to every collection of blocks using a template. In general, a randomized values template is generated using a secret key. Due to the potential vulnerability of 64-bit key to brute force attack at present, the secret key length required has to be at least 128 bits or more. Again, the long secret key can provide better security to the simple XOR operation.

Template size, secret key size and number of blocks has to be put into consideration to ensure the level of security and efficiency of operation while dealing with the macroblock data. Although we realized that the size of each macroblock data is 16x16 blocks, but this size is not always suitable for encryption that comes after compression. The size of the macro-blocks data is no longer than 256 blocks and their length depend on how much the compression has done its part. In other words, the lengths of the compressed macroblocks are varied and relatively small compare to the actual size of macroblock data. To make the XOR implementation more flexible, the template size, 8x8 and 16x16 are chosen, depending on the size of compressed data. Use of 16x16 size template required fewer loops for large data while 8x8 size template is used to minimize the block padding while less data is applied. The increasing use of template probably increases the possibility of cryptanalysis attacks because the probability of finding a repeated pattern in the entire data is higher. Thus, the proper template size need to be selected depend on the entire data size. The 4x4 size is discouraged because it required more loops and ease to attack.

Macroblocks are chosen as the target for encryption due to its characteristic as the smallest unit of compressed data in MPEG. They are small and manageable. In addition, the slice layer information will remain intact if encryption is done at the macroblock level. Another reason is that as Alattar and Al-Regib had found, encrypting I-MBs gives higher security level than only encrypting I-pictures [6,7].

Another aspect that needs to take into consideration is the approach placement or assignment scheme. The

higher the level used such as picture layer, the coarser the information will be considered. Selective encryption in this layer might be vulnerable to attacks because of data exposure. On the other hand, the lower the level used such as block layer the more information will be considered, which can be very detailed but the degradation of performance might not be avoidable. Thus, macroblock level is chosen as the best approach.

Particularly, the essential component of this approach is the template. If the template contains long run of zeros or ones, the original data can be easily guessed from resulting encrypted data. In the following section, the template generation is discussed.

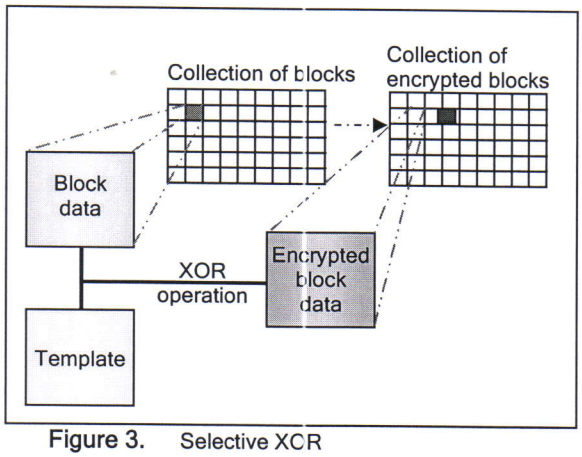

Figure 3. Selective XOR

3.1.1 Template Generation

The generation of each block value in the template is important to influence the efficiency of encryption in term of speed and security. Since the secret key is used in the generation, this template is implicitly vital to be generated such that the secret key is concealed from adverse attacks even if the template is revealed. Thus, the "strong" template that able to satisfy these requirements needs to be considered in designing the solution.

For fast execution, simple operation such as bit-wise XOR, AND, OR and NOT is considered. In term of security, template that consists of randomized values generated is considered as a secured template. In our implementation, parts of the macroblock data are utilized to achieve the "secure" template.

In chorus, the solution for secret key concealment can be as easy as using a standard symmetric encryption algorithms such as AES, Blowfish, etc. However, it is worthless if another key is needed for concealing a secret key. The solution to this is to use the secret key only once with no linear correlation between the secret key and the output values.

Our model suggests suggest that the secret keys are used only once by XOR-ing with the partial of the collection of blocks which have the equivalent size. Subsequently, the resulting data is AND-ed with another partial of the same collection of blocks. The XOR and AND operation are swapped intermittently until all the blocks in the collection of blocks are exercised. All resulting data are gathered and prearranged using mod function with the secret key to form the template.

The prearranged formula is described as
row =unit;
do
x = i th key mod (row+1); row--;
until row=1;
where
i th key = the number of bytes of the secret key;
unit = size of template;
row = number of row in the template;
x = number of output selected for current row.

For instance, let secret key = 11100001 01111010 0…, and unit=16. For row=15, initially convert binary to decimal number, 2nd byte of the secret key is 01111010 =122. Subsequently, using the equation mentioned, 122 mod (15+1) = 10. As the result, output 10 is selected to be used as the 15th row of data in the template. In order to accomplish the arrangement into a complete template, 16 operations are required in this case.

3.2 Macroblock Permutation

Instead of scrambling the stream in byte-wise fashion, we suggest that permutation method is conducted in block size. Basically, the entire macroblocks are shifted using a permutation box (P-box), which consists of four basic shift operations: Shift Left, Shift Up, Shift Right and Shift Down operations and an odd and even-numbered swapping routine. The secret key is used to determine the position movement of those shift operations.

As the 128-bit key length is used, this key is divided into four fractions for four types of the shift operations; each shift operation has 32 bits data to work with. However, applying 32 bits data at once in the operation seems a waste of data because the shifting process is going to be repeated since the data stream is shifted circularly. Moreover, the inefficiency increased if the picture size is small. Thus, the rational figure frequency is set, which is 4 bits for each shift operation. We can summarize that this permutation can be performed in four rounds, each round use each shift operation once for odd-numbered rows and columns, and each shift operation once for even-numbered rows and columns.

The macroblock permutation flow chart is depicted as Figure 4. By using this permutation with each shift operation in one position movement for odd and even-

numbered rows and columns once, the encrypted result is obtained. We can perceive that the original pattern is invisible after the macroblock permutation.

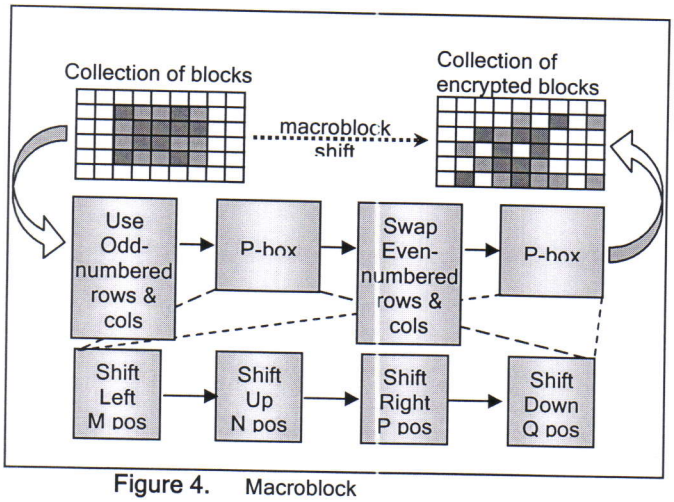

Figure 4. Macroblock

3.3 Hybrid Approach

Regarding Claude Shannon, diffusion and confusion are the two methods to counter statistical cryptanalysis [10]. The statistical structure of the plaintext is disintegrated into long range of statistics of the ciphertext in diffusion. This can be achieved by repetitively performing some permutation followed by applying a function to that permutation. The method described is similar to the execution of macroblock permutation plus a XOR template method onto the permuted macroblocks.

Alternatively, confusion make the relationship between the statistics of ciphertext and the value of encryption key as complex as possible. The intention is to make the deduction key difficult by producing the ciphertext using the key in a complex way. This notion is similar to the way of XOR template is generated where the key is XOR-ed with the macroblock data to produce the "strong" template for each GOP.

The overall diagram of the hybrid approach is shown in Figure 5. With the combination of both the proposed approaches, we believed the essence of the desired attributes, diffusion and confusion are seized.

4 Results

Experiment has been carried out to examine the performance based on the encryption and decryption execution time, and the size of the MPEG file before and after encryption is performed with the proposed approaches.

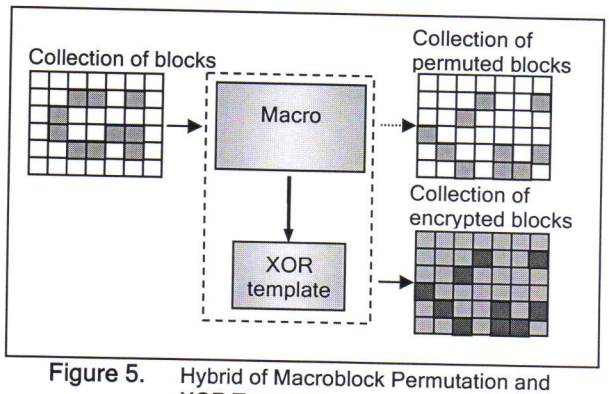

Figure 5. Hybrid of Macroblock Permutation and XOR Template.

The parameters for tuning the performance are the block size used and the encryption method applied. The sample used with some of its properties is stated as followed:

Sample1: ts.mpg (downloaded from [11])
File size (30KB), 352x240x3, 10 frames.

Macroblock permutation and XOR template are two implemented methods which are written in MATLAB. The implementation involves the creation of a permutation list and a blocked template from a given secret key. Additionally, another program is written to extract and manipulate the macroblock data from the MPEG file for encryption and decryption simulation. They are used to produce the results shown below. As the methods are not integrated in the actual MPEG decoder, the result does not consider the delay or constraints that might produce due to the integration. In contrast, the data simulation program might not be as efficient as the MPEG decoder.

Table 2. Result Of Encryption And Decryption Using 16x16 Blocks

	Execution time (s) / File size(KB)	
	Encryption	Decryption
Permute	1.76 / 31.8	1.09 / 29.3
XOR	1.37 / 34.2	1.32 / 29.3
Hybrid	1.82 / 34.4	1.32 / 29.3

Table 3. Result Of Encryption And Decryption Using 8x8 Blocks

	Execution time (s) / File size(KB)	
	Encryption	Decryption
Permute	2.41 / 30.4	0.99 / 29.3
XOR	1.32 / 30.6	1.21 / 29.3
Hybrid	2.58 / 31.1	1.21 / 29.3

Note:
Permute = Macroblock permutation
XOR = XOR template
Hybrid = Macroblock permutation plus XOR template

5 Discussion

From the result shown, the discussion can be categorized into two parts. We discussed about the encryption and decryption execution time in the first part and followed by deliberation about the encrypted MPEG file size.

5.1 Encryption and Decryption Execution Time

Generally, the decryption execution time for all methods is shorter than encryption execution time. One of the reasons for this is probably because of the involvement of key generation that required additional times in the encryption process.

Hybrid approach has greater encryption execution time compared to macroblock permutation and XOR template because they work in sequential manner, permutation followed by XOR operation for each collection of blocks until the entire video content is exercised.

XOR template operation is faster than permutation in this experiment. This is only correct if the data values are averagely small because for numerous data values, it has to be served iteratively for smaller bits of stream instead of a single chunk at once due to data type limitation in MATLAB implementation.

5.2 Encrypted Stream Size

The size of the encrypted video stream increased the least from permutation method, XOR template operation to hybrid approach (the most increased in size). Basically, there are two procedures that are able to cause the increasing of stream size: encryption header information and block padding.

Encryption header information is added in each slice to allow the decryption to works. The information includes the encryption type, permutation list and/or XOR template. If the compressed macroblock data is unable to form a complete block size data, the block padding is applied. Thus, the smaller block used the less increase on the size of the encrypted file. However, there is the tradeoff between the execution time and the encrypted file size. Longer time is needed to process the entire data.

6 Conclusion

The proposed method which uses the macroblock permutation and XOR template is believed to provide secure and fast implementation. The security is achieved through the use of both the confusion and diffusion found on the permutation and the XOR template. The faster speed is gained by using the proposed method which eliminated the use of the costly contemporary encryption/decryption algorithm.

7 References

[1] MPEG.ORG website. http://www.mpeg.org/MPEG/

[2] C-Cube Microsystems Inc, *Compression Technology: an MPEG Overview*, Sept. 1999.

[3] Y. Li, Z. Chen, S. Tan and R. Campbell. *Security Enhanced MPEG Player.* In Proceeding of the IEEE 1st International Workshop on Multimedia Software Development (MMSD'96), Berlin, Germany, March 1996.

[4] Shi and Bhargava *An Efficient MPEG Video Encryption Algorithm.* Proceedings of the 1998 IEEE Symposium on Reliable Distributed Systems, IEEE Comp. Soc. Los Almitos, CA, USA 98CB36281 P381-386, 1998.

[5] Qiao and Nahrstedt *Comparison of MPEG Encryption Algorithms.* International Journal of Computers and Graphics, special issue: "Data Security in Image Communication and Network" vol.22 , January 1998

[6] Alattar, A.M. and Al-Regib, G.I., *Evaluation of selective encryption techniques for secure transmission of MPEG-compressed bit-streams.* in Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 340-343, 1999.

[7] Alattar, A.M. and Al-Regib, G.I., *Improved selective encryption techniques for secure transmission of MPEG-compressed bit-streams.* in Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999.

[8]L.Tang *Method for Encrypting and Decrypting MPEG Video Data Efficiently.* In Proceedings of the Fourth ACM International Multimedia Conference (ACM Multimedia '96), pages 219-230, Bosten, MA, November 1996.

[9] Qiao and Nahrstedt *A New Algorithm for MPEG Video Encryption.* In Proceedings of the 1st International Conference on Imaging Science, Systems and Technology (CISST'97), page 21-29, Las Vegas, Nevada., July 1997.

[10] William Stallings, *Cryptography and Network Security Principles and Practices*, Third Edition. page 66-67, 2003.

[11] Berkeley Multimedia Research Center (BMRC) website. http://bmrc.berkeley.edu/frame/ research /mpeg/

[12] The MPEG Homepage. http://mpeg.telecomitalialab.com