An Overview of Video Encryption Techniques

Article · February 2010

DOI: 10.7763/IJCTE.2010.V2.123

CITATIONS
28

READS
201

3 authors, including:

Mohamed Abomhara
Universitetet i Agder
19 PUBLICATIONS **269** CITATIONS

SEE PROFILE

Othman O. Khalifa
International Islamic University Malaysia
415 PUBLICATIONS **1,668** CITATIONS

SEE PROFILE

International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010
1793-8201

An Overview of Video Encryption Techniques

M. Abomhara, Omar Zakaria, Othman O. Khalifa

***Abstract*- With the rapid development of various multimedia technologies, more and more multimedia data are generated and transmitted in the medical, commercial, and military fields, which may include some sensitive information which should not be accessed by or can only be partially exposed to the general users. Therefore, security and privacy has become an important. Over the last few years several encryption algorithms have applied to secure video transmission. While a large number of multimedia encryption schemes have been proposed in the literature and some have been used in real products, cryptanalytic work has shown the existence of security problems and other weaknesses in most of the proposed multimedia encryption schemes. In this paper, a description and comparison between encryption methods and representative video algorithms were presented. With respect not only to their encryption speed but also their security level and stream size. A trade-off between quality of video streaming and choice of encryption algorithm were shown. Achieving an efficiency, flexibility and security is a challenge of researchers.**

***Keywords: Video* encryption, video transmission, Video Coding, Scalable Video Streaming.**

I INTRODUCTION

Cryptography is the art and science of protecting information from undesirable individuals by converting it into a form non-recognizable by its attackers while stored and transmitted. Data cryptography mainly is the scrambling of the content of data, such as text, image, audio, video and so forth to make the data unreadable, invisible or unintelligible during transmission or storage called Encryption. The main goal of cryptography is keeping data secure form unauthorized attackers. The reverse of data encryption is data Decryption, which recuperate the original data. Since cryptography first known usage in ancient Egypt it has passed through different stages and was affected by any major event that affected the way people handled information. In the World War II for instance cryptography played an important role and was a key element that gave the allied forces the upper hand, and enables them to win the war sooner, when they were able to dissolve the Enigma cipher machine which the Germans used to encrypt their military secret communications [1].

Manuscript received June 20, 2009

Mohamed Abomhara, - Masters Student, Department of Computer Science & Information Technology, University Malaya, Kuala Lumpur, Malaysia, and phone: +60172471620 Email: m.abomhara@gmail.com.

Dr.Omar Zakaria - Senior lecture, Department of Computer Science & Information Technology, University Malaya, Kuala Lumpur, Malaysia,omar@fsktm.um.edu.my.

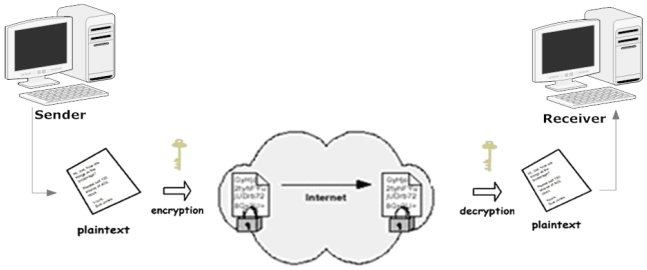
Dr. Othman O. Khalifa - Head of the department of Electrical and Computer Engineering, International Islamic University Malaysia., Kuala Lumpur, Malaysia,

In modern days cryptography is no longer limited to secure sensitive military information but recognized as one of the major components of the security policy of any organization and considered industry standard for providing information security, trust, controlling access to resources, and electronic financial transactions. The original data that to be transmitted or stored is called plaintext, the one that can be readable and understandable either by a person or by a computer. Whereas the disguised data so-called ciphertext, which is unreadable, neither human nor machine can properly process it until it is decrypted. A system or product that provides encryption and decryption is called cryptosystem. Cryptosystem uses an encryption algorithms which determines how simple or complex the encryption process will be, the necessary software component, and the key (usually a long string of bits), which works with the algorithm to encrypt and decrypt the data [2] [3]. In the 19th century, a famous theory about the security principle of any encryption system has been proposed by Kerchhoff. This theory has become the most important principle in designing a cryptosystem for researchers and engineers. Kirchhoff observed that the encryption algorithms are supposed to be known to the opponents. Thus, the security of an encryption system should rely on the secrecy of the encryption/decryption key instead of the encryption algorithm itself. For even though in the very beginning the opponent doesn't know the algorithm, the encryption system will not be able to protect the ciphertext once the algorithm is broken. The security level of an encryption algorithm is measured by the size of its key space [3]. The larger size of the key space is, the more time the attacker needs to do the exhaustive search of the key space, and thus the higher the security level is. In encryption, the key is piece of information (value of comprise a large sequence of random bits) which specifies the particular transformation of plaintext to ciphertext, or vice versa during decryption. Encryption key based on the keyspace, which is the range of the values that can be used to assemble a key. The larger keyspace the more possible keys can be constructed (e.g. today we commonly use key sizes of 128,192,or 256 bit , so the key size of 256 would provide a 2256 keyspace) [3][4]. The strength of the encryption algorithm relies on the secrecy of the key, length of the key, the initialization vector, and how they all work together. Depend on the algorithm, and length of the key, the strength of encryption can be considered. Assume that if the key can be broken in three hours using Pentium 4 processor the cipher consider is not strong at all, but if the key can broken with thousand of multiprocessing systems within a million years, then the cipher is pretty darn strong. There are two encryption/decryption key types: In some of encryption

technologies when two end points need to communicate with one another via encryption, they must use the same algorithm, and in the most of the time the same key, and in other encryption technologies, they must use different but related keys for encryption and decryption purposes. Cryptography algorithms are either symmetric algorithms, which use symmetric keys (also called secret keys), or asymmetric algorithms, which use asymmetric keys (also called public and private keys).

A. *Symmetric key Algorithms*

In symmetric key encryption, the sender and receiver use the same key for encryption and decryption. As shown in figure 1. symmetric key encryption is also called secret key, because both sender and receiver have to keep the key secret and properly protected[4][5] Basically, the security level of the symmetric keys encryption method is totally depend on how well the users keep the keys protected. If the key is known by an intruder, then all data encrypted with that key can be decrypted.



II Symmetric key Algorithms

This is what makes it more complicated how symmetric keys are practically shared and updated when necessary. Symmetric keys can provide confidentiality but they can not provide authentication, because there is no way to prove through cryptography who actually sent a massage if two people are using the same key. Due to that, with all the problem and defects that symmetric keys have they still used in many applications, because they are so fast and can be hard to break if using a large key size. Symmetric keys can handle a large amount of data that would take an unacceptable amount of time with asymmetric keys to encrypt and decrypt.

The most popular symmetric key algorithms are Data Encryption Standard (DES), Triple DES, and Advance Encryption.

1) The Data Encryption Standard (DES)

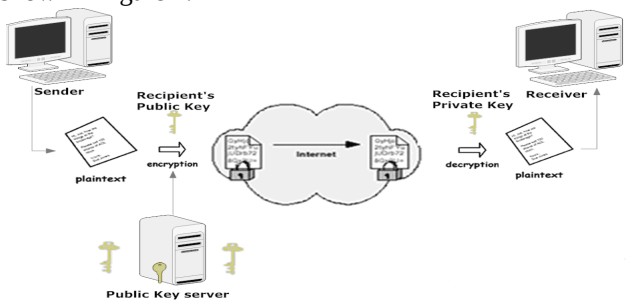
DES is one of the most important examples of a block cipher. DES was the result of a contest set by the U.S. National Bureau of Standards (now called the NIST) in 1973, and adopted as standard applications in 1977. The winning standard was developed at IBM, as a modification of the previous system called LUCIFER. The DES is widely used for encryption of PIN numbers, bank transactions, and the like. The DES is an example of a block cipher, which operates on blocks of 64 bits at a time, with an input key of 64 bits. Every 8th bit in the input key is a parity check bit which means that in fact the key size is effectively reduced to 56 bits[6][7].

2) Advance Encryption Standard (AES)

In 1997, the NIST called for submissions for a new standard to replace the aging DES. The contest terminated in November 2001with the selection of the Rijndael cryptosystem as the Advanced Encryption Standard (AES) [4] [8]. The Rijndael cryptosystem operates on 128-bit blocks, arranged as 4×4 matrices with 8-bit entries. The algorithm can use a variable block length and key length; the latest specification allowed any combination of keys lengths of 128, 192, or 256 bits and blocks of length 128, 192, or 256 bits

B. *Asymmetric key Algorithms*

Asymmetric key algorithm is also called public key algorithm. Public Key Cryptography was first described publicly by Stanford University professor Martin Hellman and graduate student Whitfield Diffie in 1976[9]. They described a two-key crypto system in which two parties could securely communicate over a non-secure communications channel without having to share a secret key and address the problem of secret key distribution by using two keys instead of a single key . In public key algorithm there are two keys are used. A public key, which can be known by everyone, and a privet key, which should be kept secret known only by the owner. As shown in figure 2.



III Asymmetric key Algorithms

If the massage is encrypted by one key the other key is required in order to decrypt the message [4]. The public key and private key are mathematically related. However it does not mean that, if someone got the public key he/she will be able to figure out the private key , but if someone got the private key, then here is big trouble, because private key should be accessed only by the owner no one else [4][5].

On the condition that the authentication is required, the data would be encrypted with the sender's private key then each person has the corresponding public key will be able to decrypt the data. This provides a confidence to the receiver that the data has been encrypted by one who has the possession of that private key. Encrypting data with private key is called *open message format*, since confidentiality is not ensured. Anyone with a copy of the corresponding public key can decrypt the data. The most popular asymmetric key algorithms are Rivest- Shamir Adelman (RSA).

1) Rivest- Shamir Adelman

RSA is one of the most used public key algorithms today. This algorithm was invented in 1977 by Ron Rivest, Adi Shamir, and Len Adelman. The RSA is based on the idea of factorization of integers into their prime. Assume that Alice and Bob want to communicate with one other. Bob chooses

two distinct large primes p and q and multiplies them together to form N, N = p*q. He also chooses an encryption exponent e, such that the, greatest common divisor of e and [(p-1)*(q-1)] is 1. That is gcd(e,[(p-1)*(q-1)])=1. He computes his decryption key d, d=1/e (mod [(p-1)*(q-1)]). Now he makes the pair (N,e) public and keeps p and q secret. This how to Generating keys, Encryption and decryption are of the following form, for some plain text block M and ciphertext block C: C=Me mod n, M= Cdmod n = (Me) mod n = Medmod n Both sender and receiver must know the values of n and e, and only the receiver knows the value of d. this make a public key encryption of KU = {e,n} and private of KR {d,n}.

II Symmetric key Algorithms vs. Asymmetric key Algorithms

Asymmetric algorithms work much slowly then the symmetric algorithm, because they use more complex mathematics to perform their functions, which require more processing time. With public key, you can just send out your public key to all of the people whom you need to communicate with, instead of keeping track of a unique for each one of them. As show in table 1 the following list of the advantage and disadvantage of symmetric and Asymmetric key systems.

- Symmetric key is much faster than asymmetric systems. On the other side asymmetric key Works much more slowly.
- In symmetric key the security is dependent on the length of the key, if using a large key size the algorithm will be hard to break, because symmetric algorithms carry out relatively simplistic mathematical functions on the bits during the encryption and decryption processes.
- Symmetric key requires a secure mechanism to deliver keys properly. While, asymmetric key provide a better key distribution than symmetric systems.
- Symmetric key provides confidentiality but not authenticity, because the secret key is shard. However, asymmetric key can provide authentication and confidentiality.

In symmetric key Each pair of users needs a unique key, if a user has N trading partners, then N secret keys must be maintained so as the number of individuals increases, so does the number of keys. However, management of the symmetric keys becomes problematic

III Digital video encryption scheme

With digital video transmission, encryption methodologies are needed that can protect digital video from attacks during transmission. Due to the huge size of digital videos, they are usually transmitted in compressed formats such as MPEG [10], or H.264/AVC [11]. Thus, the encryption algorithms for digital video are usually working in the compressed domain. Several encryption algorithms to secure video streaming have been proposed. Most of them tried to optimize the encryption process with respect to the encryption speed, and display process.

A. *Naïve Algorithm*

The most straight-forward method to encrypt every byte in the whole Moving Picture Experts Group (MPEG) (ref website) stream using standard encryption schemes such as DES or AES. The idea of Naïve algorithm to treat the MPEG bit-stream as text data and does not use any of the special structure [12][13][14] . Naïve algorithm ensures the security level to the entire MPEG stream by standard encryption schemes because no effective algorithm to break encryption schemes especially AES nor triple DES so far. However, this algorithm not applicable solution for big video, because it is very slow especially when we use triple DES. Because of the encryption operation the delay increases and overload will be unacceptable for real time video encryption.

B. *Pure Permutation Algorithm*

The idea of pure permutation algorithm is simply scrambles the bytes within a frame of MPEG stream by permutation. It is extremely useful in situation where the hardware decodes the video, but decryption must be done in software. Adam J. slagell in [15] demonstrates that the pure permutation algorithm is vulnerable to known-plaintext attack, and hence its use should be careful considered, because by comparing the ciphertext with the known frames, the adversary could easily figure out the secret permutation list. Once the permutation list is figured out, all frames could be easily decrypted. Notice that knowing one I frame of MPEG stream is enough to decrypt the permutation list based on Shannon's Theorem.

C. *Zig-Zag Permutation Algorithm*

The main idea of Zig-Zag permutation approach [16] is instead of mapping the 8x8 block to 1x64 vector in "Zig-zag" order, it maps the individual 8x8 block to a 1x64 vector by using a random permutation list (secret key). Zig-Zag permutation algorithm consists of three main steps:

- Generate a list of 64 permutations.
- Complete splitting procedure. Assume that DC coefficient is denoted by 8 digit binary numbers d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 then it is split into two numbers d_7 d_6 d_5 d_4 and d_3 d_2 d_1 d_0. Then the number of d_7 d_6 d_5 d_4 placed to DC coefficient and the number of d_3 d_2 d_1 d_0 placed to AC coefficient. Splitting procedure base on the following observations 1) the value of DC coefficient is much larger then the value AC coefficient. 2) after splitting, extra space need for a store one of the splitting number, and this will increase the size of MPEG stream. However notice that the last AC coefficient is the least important value in block which can be set zero with no significant visual degradation.
- Apply the random permutation to the split block.

Since mapping Zig-Zag order and mapping according to the random permutation list have the same computational complexity, the encryption and decryption add very little overhead to the video compression and decompression processes. However this method decrease the video compression rate because the random permutation distort the probability

distribution of Discrete Cosine Transform (DCT) coefficients and make the Huffman table used less than optimal. L.Qiao and Nahrstedt in 1998 introduced two types of attacks on Zig-Zag permutation are introduced, a ciphertext only attack, and a known-plaintext attack.

The Zig-Zag permutation algorithm is vulnerable to the ciphertext only attack, the attack relies on the fact of statistical properties of the DCT coeffient, where none-zero AC coefficients are gathered in the upper left corner of the I-block. Statistical analysis by L.Qiao and Nahrstedt in 1998 which the count the number of none-zero ACs and DC coefficients from all blocks in an I-frame showed that following observation:

- DC coefficients always have the highest frequency of none-zero occurrences.
- The frequency of AC1 and AC2 are among the top 6.
- The frequency of AC3 to AC5 is among top 10.

The second problem is that, the Zig-Zag permutation algorithm can not withstand the known-plaintext attack. Assume that we know certain frames of the video in advance (known-plaintext) the secret key could be easily figured out by simply comparing the known plaintext with the corresponding encrypted frame. To solve this problem a proposed method called binary coin flipping sequence method together with two different permutation lists. For each 8x8 block, a coin is flipped. If it is a tail, the permutation list 1 (key1) is applied to the block. If it is a head, the permutation list 2 (key 2) is applied to the block. This method is vulnerable to known-plaintext attack as well, because none-zero AC coefficients have the tendency to gather in the upper left corner of the block, it would be easy for an adversary to determine which key is used.

D. *Video Encryption algorithm (VEA)*

Qiao and Nahrstedt in [17] suggested a new video encryption algorithm called VEA. Video encryption algorithm upon the statistical properties of MPEG video standard and symmetric key algorithm standard to reduce the amount of data that is encrypted. AVE is actually dividing the input video stream into chunks (a1, a2, a3, a4, … , a2n-1 ,a2n). The chunk is divided into two data segments odd list (a1, a3, a5, …. , a2n-1) and Even list (a2, a4,a6, … , a2n) , afterward the encryption key would be applied to the list even list E(a2, a4,a6, … , a2n), where E denotes an encryption function. Then the final ciphertext is a concatenation of output of encryption algorithm XOR with the odd list streams. Thus AVE algorithm is immune from known-plaintext attack, because the key will be changed for each frame(s).

E. *Video Encryption Algorithm (VEA)*

Bharagava, Shi, and Wang in [18] [19] have introduced four different video encryption algorithms Algorithm I, Algorithm II (VEA), Algorithm III (MVEA), and Algorithm IV (RVEA).

1) Algorithm I

Algorithm I uses the permutation of Huffman codewords in I-frames. This method incorporates encryption and compression in one step. The secret part of the algorithm is a permutation p which is used to permute standard MPEG Huffman codeword list. In order to save compression ratio, the permutation p must be such that it only permutes the codewords with same number of bits. Daniel Socek, and el in [20] showed that the Algorithm I is highly vulnerable to both known plaintext attack, and ciphertext-only attack. If some of video frames known in advance the adversary could easily figure out and reconstruct the secret permutation p by comparing the known frames with the encrypted frames. However, algorithm I is also subject to ciphertext-only attack, in [21] defined the low frequency error attack on algorithm I ciphertext. Basically, since permutation p is of the special form; i.e., it only shuffles codewords with the same length, the most security comes from shuffling 16bit codewords in the AC coefficient entropy table. However, since there are very limited numbers of codewords with length of less than 16bits, it is very easy to reconstruct all of the DC coefficients and most frequent AC coefficients (since these will be encoded with less than 16bit codewords). In other words, the only hard part would be to figure out how does the permutation p shuffle the 16bit codewords. But these are appearing extremely rare, and the reconstructed video may be of almost the same quality as the original.

2) Algorithm II (VEA)

The algorithm was proposed in [18] ,since the I-blocks carry the most important information so the scheme sufficient to encrypt only the sign bite of the DC coefficients in the I-frame blocks by simply XORs sign bites of DC coefficients with a secret key. The security level of this scheme depends on the length of the key. However, too long key size may be infeasible and impractical. On the other hand with a short key size, the system could be easily broken.

3) Algorithm III (MVEA)

Bharagava and Shi in [22]have made an improvement to the Algorithm II (VEA). Instead of encrypt only the sign bite of DC coefficient in the I-frame block, the sign bite of the differential values of DC coefficient and motion vectors in P-frames and B-frames can is encrypted by XORs them with the secret key. However this type of improvement makes the video playback more random and more unviewable. Just like the Algorithm II (VEA), the Algorithm III (MVEA) is relies on the secret key size.

4) Algorithm IV (RVEA)

Algorithm IV (RVEA) was proposed by Baraga, and el [19]. The different between Algorithm IV (RVEA) and Algorithm III (MVEA) is that Algorithm IV (RVEA) uses a traditional symmetric key cryptography to encrypt the sign bite of DCT coefficient and the sign bite of motion vectors. The algorithm speeds up the process of encryption by only encrypt certain sign bite in MPEG stream. Therefore, it is much better then the previous three algorithms Algorithm I, Algorithm II (VEA), and Algorithm III (MVEA) in terms of security. Furthermore, it saves up to 90% of the computation time comparing with Na ve approach.

F. *Selective Encryption Algorithm*

In order to reduce the amount of processing overhead

and to meet the security for real time video applications, selective encryption technique have been proposed. The idea of this scheme is to encrypt different levels of selective parts of MPEG stream by using the feature of MPEG layered structures (e.g. encrypting all headers and I frames, encrypting all I frames and all I blocks in P and B frames). The basic selective encryption is based on the MPEG I-frame, P-frame, and B-frame structure. It encrypts the I-frame only because, conceptually P- and B- frame are useless without knowing the corresponding I-frame.

1) AEGIS, (Encrypt I-Frame Only)

Maples and Spanos in [23][24] have introduced a new secure MPEG video mechanism called Aegis. Aegis method encrypts only the I-frame of all MPEG groups of frames in MPEG video stream, while B-frame and P-frame are left unencrypted. In additional, to make the MPEG video stream more secure Aegis also encrypts the sequence header which contains all of the decoding initialization parameters such as the picture width, height, frame rate, bit rate, and buffer size. Encrypting of the sequence header makes the MPEG identity of stream concealed and it makes the MPEG video stream unrecognizable. Finally, Aegis also encrypts the IOS end code (last 32 bits of MPEG stream) as a result to further conceal the bit stream of MPEG identity. Aegis uses the DES encryption engine for encryption process. Iskender Agi and Li Gong in [25] have shown that encryption I-frame alone may not be sufficiently secure for some type of video. Their experiment showed that it is possible to know some scenes from the decoded P-frame and B-frame. Iskender Agi and Gong experimented with encrypting all of I-block in the P-frames and B-frames in addition to encrypt all I-frame the security level have been significantly increased. They have also suggested that to increase the frequency of I frames to enhance the security. It has the main drawback of increasing the length of string and consequentially the encryption time. Finally, it is important to notice that this type of security is not particularly good for applications where the security is one of the top priorities (such as the important military or business teleconferencing) but it might be sufficient as a quality degradation of the entertainment videos, such as the pay TV broadcast.

2) Sign-Bit of DCT Coefficients

Shi and Bharagava [18] used a secret key to transform the sign bits of the DCT coefficients of MPEG video data. The secret key (k1, k2, k3,….., k2m) is randomly generated with length 2m, where the number of key and the length of key is not limited. If the sign bits of DC and AC coefficients are represented by S, (s1, s2, s3,…. , s2m), then the encrypted data is Ek(Si) = bi xor si of length 2m. The encryption operation randomly changes the sign bits of DCT coefficients. The decryption function Ek-1 is the same as the encryption function since Ek (Ek) = S. For a key of length m an adversary needs to try 2m times in order to find a key. In this algorithm, several keys can be used to enhance the security. For example, in the 2 keys scheme, one key is for Y blocks and the other for Cb and Cr blocks. In the 3 keys scheme, one for I frames, one for B frames, and one for P frames.

Headers, Lookabaugh et al [26] proposed the selective encryption of MPEG-2 video, the one that used in most current digital television applications. They use the fact that the typical high performance MPEG-2 encoded bitstreams only uses a small portion of bits in important headers (video sequence, group of pictures, picture, and slice). It can be simple to vague such headers because of a usual practice in encoding of aligning these headers and the multiplex level at which encryption is performed. However, fields in such headers can be quite defenseless to attack, even if obscured by selective encryption, for a variety of reasons: the fields are often static, they can be guessed from external information that is probably available to an attacker, they can be guessed from other information in the bitstream (e.g., picture type can be guessed from picture size, an example of the cryptanalytic technique of traffic analysis), or they can be ignored, albeit with nontrivial consequences for decoded image quality. They evaluated each of these fields, and proposed and tested attacks. For example, they showed that a perceptual attack on the quantizer-scale-code syntactic element is feasible albeit with nontrivial picture degradation: in typical sequences there is a strongly peaked distribution for this code, and a perceptual attack would be to always use an expected value for this code in place of the correct value. It is clearly that the resulting reconstruction is distorted, but it is not obvious that it is sufficiently distorted to cause a pirate to pay for a clean version if the distorted version is available for free. A more encouraging example is the choice of the macroblock-type field that signals to a decoder the type of prediction used for each macroblock (16-pixel vertical by 16-pixel horizontal region) in a video frame. Although this field does not use a very large fraction of the bitstream (on the order of a couple of percent typically), if absent it is very difficult for a decoder to guess it and to decode remaining material correctly (since the macroblock type uses a Huffman code and, if incorrectly decoded, a decoder has a hard time resynchronizing) [27].

3) Byte-Encryption

Griswold et al. in [28][29] have proposed to randomly encrypt bytes in an MPEG stream for free distribution, while the original bytes at the corresponding positions are transferred in encrypted form to legitimate users. This is actually equivalent to encrypting byte at random positions. The authors find that encrypting 1% of the data is sufficient to make a video undecodable or at least invisible. However, the cryptanalysis given is entirely insufficient. Consider the worst case where only MPEG header data is encrypted by chance using this approach. It is well known that header data may be reconstructed easily provided the encoder in use is known. Additionally, no attack scenario is considered but only the case of playing the protected video in a standard decoder is covered. In order to guarantee a certain level of security, a higher amount of bytes need to be encrypted and care needs to be taken about which bytes are encrypted. J. Wen at al. [30] describes a more general approach as part of the MPEG 4 IPMP standard, named Syntax Unware Runlength-based

Selective Encryption (SURSLE). This algorithm encrypt X bits, the next Y bits are left in plain-text; the next Z bits encrypted again, and so on. In addition to the above mentioned security problems, both schemes partially destroy the MPEG bitstream syntax and potentially emulate important MPEG markers causing a decoder to crash.

IV Comparisons of Video Encryption Algorithms

We have currently known encryption algorithms for secure video streams and evaluated them with respect to three metrics: security level, encryption speed, and encrypted MPEG stream size. As our summary shown in Table 1, Naïve Algorithm and Video Encryption Algorithm (VEA) are the most secure algorithms, where Zig-Zag Permutation Algorithm has serious security flaws and can not withhold the known plaintext attack nor the ciphertext only attack. With respect to encryption speed, Pure Permutation Algorithm and Zig-Zag Permutation Algorithm are very fast, and Naïve Algorithm is very slow due to applying DES on whole MPEG stream. When comparing the algorithms in terms of size metric, VEA, Pure Permutation Algorithm and Naïve Algorithm do not change their size, which is very much desirable. On the other hand, Zig-Zag Permutation Algorithm significantly increase the stream size which defeats the compression purpose of MPEG encoding. In summary, there are trades offs when applying different encryption algorithms to MPEG encoded video and its choice depends on the applications. We believe that VEA meets the requirements of most multimedia applications because it provides overall high security, size preservation, and relatively fast encryption. Any other algorithm suffers from either low security, or low speed, or stream size increases.

V Conclusion

In this paper a surveyed of the currently known methods of cryptography were presented. The two different types of the encryption methods (Symmetric key encryption and Asymmetric key encryption) were highlighted and evaluated with respect to their security level and encryption speed. A discussed about the advantages and disadvantages of each of them. Moreover, currently known video encryption algorithms for video streams were described. We have showed, Naïve algorithm and video encryption algorithm are the most secure algorithms, where zig-zag permutation algorithm has serious security flaws and con not withhold the known-plaintext-attack nor the ciphertext-only attack. . with the respect to encryption speed, pure permutation algorithm and zig-zag permutation algorithm are very fast, and Naïve algorithm is very slow specially while applying DES on whole video. In summary, there is a trade-offs when applying different encryption algorithms to MPEG video stream and its choice rely on the applications.

Acknowledgement

Thanks in advance for the entire worker in this project, and the people who support in any way, also I want to thank UM and IIUM for the support they offered.

References

[1] Kahn, David , (1980). Cryptology Goes Public, Communications Magazine, IEEE, available from: http://ieeexplore.ieee.org/iel5/35/23736/01090200.pdf?tp= &isnumber=&arnumber=1090200. (Accessed December 28, 2008).
[2] Kessler , Gary C., (1998). An Overview of Cryptography, available from: http://www.garykessler.net/library/crypto.html#intro. (Accessed December 28, 2008).
[3] B. White, Gregory, (2003). Cisco Security+ Certification: Exam Guide, McGraw-Hill.
[4] shon harris, (2007). SICCP Exam Guide, fourth edition, McGraw-Hall
[5] Stallings, William, (2007). Network Security Essentials, applications and Standards , Pearson Education, Inch.
[6] Wayne G. Barker, "Introduction to the analysis of the Data Encryption Standard (DES)", A cryptograph-ic series, Vol. 55, p. viii + 190, Aegean Park Press, 1991.
[7] Kofahi, N.A., Turki Al-Somani, Khalid Al-Zamil. "Performance evaluation of three encryption/decryption algorithms" 2005 IEEE International Symposium on Micro-NanoMechatronics and Human Science, Publication Date: 30-30 Dec. 2003. Volume: 2, pp 790-793.
[8] Jean-Yves chouinard.,. Design of secure computer systems CSI4138/CEG4394 notes on the advanced encryprion standard (AES), available from http://www.site.uottawa.ca/~chouinar/Handout_CSI4138_AES_200 .pd f. (Accessed February 15,2009).
[9] Diffie , Whitfield & Hellman, Martin E, (1976) . New Directions In Cryptography, IEEE TRANSACTIONS ON INFORMATION THEORY, available from: http://www-ee.stanford.edu/~hellman/publications/24.pdf. (Accessed on Decemper 28, 2008).
[10] MPEG Technology Group, \http://www.chiariglione.org/mpeg/," (Accessed on March 2, 2009).
[11] Ostermann, J., Bormans, J., List, P., Marpe, D., Narroschke, M., Pereira, F., Stockhammer, T., Wedi, T. "Video coding with H.264/AVC: tools, performance, and complexity. IEEE circuits and system magazine , Vol 4,issue 1 , pp. 7-28, 2004.
[12] Salah Aly. A Light-Weight Encrypting For Real Time Video Transmission. Available from http://www.cdm.depaul.edu/research/Documents/TechnicalReports/200 4/TR04-002.pdf. (Accessed on March 2, 2009).
[13] S. Lian, Multimedia Content Encryption: Techniques and Applications. CRC, 2008.
[14] C.-P. Wu, C.-C. J. Kuo, "Design of integrated multimedia compression and encryption systems," IEEE Trans. Multimedia, vol. 7, no. 5, pp. 828-839, 2005.
[15] Adam J. Slagell. Known-Plaintext Attack Against a Permutation Based VideoEncryption Algorithm. Available from http://eprint.iacr.org/2004/011.pdf .(Accessed on March 2, 2009).
[16] L. Tang, For encrypting and decrypting MPEG video data efficiently," in Proceedings of The Fourth ACM International Multimedia Conference (ACM Multimedia'96), (Bosten, MA), pp. 219{230, November 1996.
[17] L. Qiao and K. Nahrstedt, \A new algorithm for MPEG video encryption," in Proceedings of The First International Conference on Imaging Science, Systems, and Technology (CISST'97), (Las Vegas, Nevada), pp. 21{29, July 1997.
[18] C. Shi and B. Bhargava, "A Fast MPEG Video Encryption Algorithm," Proceedings of the 6th International Multimedia Conference, Bristol, UK, September 12-16, 1998.
[19] C. Shi, S.-Y. Wang and B. Bhargava, "MPEG Video Encryption in Real-Time Using Secret key Cryptography," 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas, NV, June 28 - July 1, 1999.
[20] T. Seidel, D. Socek, and M. Sramka, "Cryptanalysis of Video Encryption Algorithms ," to appear in Proceedings of The 3rd Central European Conference on Cryptology TATRACRYPT 2003, Bratislava, Slovak Republic, 2003.
[21] T. Seidel, D. Socek, and M. Sramka, "Cryptanalysis of Video Encryption Algorithms ," to appear in Proceedings of The 3rd Central European Conference on Cryptology TATRACRYPT 2003, Bratislava, Slovak Republic, 2003.
[22] B. Bhargava and C. Shi, "An Efficient MPEG Video Encryption Algorithm", IEEE Proceedings of the 17th Symposium on Reliable Distributed Systems, 1998, Pages 381 – 386.
[23] T.B. Maples and G.A. Spanos, "Performance study of selective encryption scheme for the security of networked real-time video," in Proceedings of the 4th International Conference on Computer and Communications, Las Vegas, NV, 1995.

[24] G.A. Spanos and T.B. Maples, "Security for Real-Time MPEG Compressed Video in Distributed Multimedia Applications," in Conference on Computers and Communications, 1996, pp. 72-78.
[25] Iskender. Agi and L. Gong, \An emprical study of MPEG video transmissions," in Proceedings of The Internet Society Symposium on Network and Distributed System Security, (San Diego, CA), pp. 137{144, Febuary 1996.
[26] T. Lookabaugh et al., \Selective encryption of MPEG-2 video," in Proceedings of the SPIE Multimedia Systems and Applications VI, (Orlando, FL), September 2003.
[27] T. Lookabaugh and D. C. Sicker, \Selective encrytion for consumer applica- tions," IEEE Communications Magazine, pp. 124{129, May 2004.
[28] C. Griwotz, \Video protection by partial content corruption," in Proceedings of Multimedia and Security Workshop at the 6th ACM International Multimedia Conference, (Bristol, England), pp. 37{39, 1998.
[29] C. Griwotz, O. Merkel, J. Dittmann, and R. Steinmetz, \Protecting vod the easier way," in Proceedings of Multimedia and Security Workshop at the 6th ACM International Multimedia Conference, (Bristol, England), pp. 21{28, 1998.
[30] J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin, \A format compliantcon ¯gurable encryption framework for access control of video," IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, pp. 545{557, June 2002.

Othman O. Khalifa received his Bachelor's degree in Electronic Engineering from the Garyounis University, Libya in 1986. He obtained his Master degree in Electronics Science Engineering and PhD in Digital Image Processing from Newcastle University, UK in 1996 and 2000 respectively. He worked in industrial for eight years and he is currently Professor and Head of the department of Electrical and Computer Engineering, International Islamic University Malaysia. His area of research interest is Communication Systems, Information theory and Coding, Digital image / video processing, coding and Compression, Wavelets, Fractal and Pattern Recognition. He published more than 130 papers in international journals and Conferences. He is SIEEE member, IEEE computer, Image processing and Communication Society member.

Mohamed Abomhara-. He completed his undergraduate degree in Computer Science at 7th October University Banewiled, Libya in 2006. He is a Master candidate in Data Communication and Computer Network at Faculty of Computer Science & Information Technology University of Malaya, Malaysia. He has done many projects on video encryption. His research interests are in computer security, encryption/decryption and video processing.

Omar Zakaria- He completed his undergraduate degree in Computer Science at the Computer Centre, University of Malaya (UM), Kuala Lumpur in 1994. He started work as an analyst programmer in Maybank Bhd at Maybank Tower, Jln Tun Perak, Kuala Lumpur in October 1994. However, he left Maybank in February 1995 because he got scholarship from UM to pursue his Master degree. He obtained his Master and PhD in information systems security management from the Royal Holloway, University of London, United Kingdom, in 1996 and 2007, respectively. He joined the University of Malaya as a tutor in February 1995 at Centre for Foundation Studies in Science. He was appointed to Lecturer in December 1996, and transferred to Faculty of Computer Science & Information Technology in April 1997. Subsequently, he was promoted to Senior Lecturer in April 2006. His research interests are in information systems security management (ISMS).

Table1 Comparative Results of the encryption algorithms

	Complexity	Speed	Memory Requirement	Key Type	Key Length	Key Space size	Security level
DES	Complex	High	N/A	Private key	56 bits, 48 bits sub-key	2^{56}	Low
AES	Complex	High	Very low	Private key	128 bite,192bits , 256 bits	2^{128}, 2^{192}, 2^{256}	High
RSA	Simple	High	N/A	Public key	Variable	Variable	High

Table2 Comparisons of Video Encryption Algorithms

Algorithms	Security	Speed	Size	Encryption Ratio
Naïve	High	Slow	No change	100%
Pure Permutation	Low	Super fast	No change	100%
Zig-Zag Permutation	Very low	Very fast	Big increase	100%
VEA	High	Fast	No change	50%
Selective	Moderate	Fast	Increase	1% -100%