

A Survey of Multimedia Security

Borko Furht

Department of Computer Science and Engineering
Florida Atlantic University, 777 Glades Road
Boca Raton, FL 33431-0991, U.S.A.
E-mail: *borko@cse.fau.edu*

Daniel Socek

Department of Computer Science and Engineering
Florida Atlantic University, 777 Glades Road
Boca Raton, FL 33431-0991, U.S.A.
E-mail: *dsocek@brain.math.fau.edu*

August 21, 2003

1 Introduction

It is evident that a new era of communications and information exchange has been underway for at least two decades. This remarkable revolution in the way we interact with one another happened almost overnight, leaving many of us unaware of the true vast and open nature of the *cyberspace*. Every day people send out love messages, sensitive personal information, financial transactions, and corporate documents, and even make business deals over the global communications channel. As the performance of the computer and network components increased in the last few years, people can now exchange complex multimedia data, speech and videos. In such a setting, one natural

question that arises is the security and confidentiality of a *digital packet* of information.

2 Basics of Cryptography

Multimedia security is based on cryptography. In fact, some basic concepts from cryptography are used as building blocks (primitives) for applications in multimedia security. For a better understanding of issues concerning security of multimedia data, an overview of cryptography is presented at first.

2.1 Definition and Goals of Cryptography

Cryptography is a study of techniques (called cryptosystems) that are used to accomplish the following four goals:

- Confidentiality
- Data Integrity
- Authentication
- Non-repudation

A study of techniques used to break existing cryptosystems is called *cryptanalysis*. Since cryptography and cryptanalysis are greatly dependent of each other, people refer to *cryptology* as a joint study of cryptography and cryptanalysis.

Let us spend some time trying to understand all four goals of cryptography. *Confidentiality* means that the communication material is confidential and that it is only accessible to the desired communicating parties. An undesired communicating party (called adversary) must not be able to access the communication material. This goal of cryptography is a basic one, one that has been always addressed and enforced throughout the history of cryptographic practice. *Data integrity* means that the communication material cannot be altered in any way. If the information is altered, all communicating parties can detect this. Means of authenticating

desired communicating parties is referred to as *authentication*. One of the main examples of authentication includes digital signatures. Finally, *non-repudation* means that the receiver can prove to everyone that sender did indeed send the message; i.e., the sender cannot claim that he or she didn't encrypt and/or sign certain digital information. Fortunately, modern cryptography has developed techniques to handle all four goals of cryptography.

Today, there are two types of cryptosystems: symmetric (private) key cryptosystems and asymmetric (public) key cryptosystems. Most people have chosen to call the first group simply *symmetric key cryptosystems*, and the popular name for the the second group is just *public key cryptosystems*.

2.2 Symmetric Key Cryptosystems

Symmetric key cryptosystems have been around for thousands of years. All classical cryptosystems (that is cryptosystems that were developed before 1970s) are examples of symmetric key cryptosystems. In addition, most modern cryptosystems are symmetric as well. Some of the most popular examples of modern symmetric key cryptosystems include AES (Advanced Encryption Standard), DES (Data Encryption Standard), IDEA, FEAL, RC5, and hundreds of others.

All symmetric key cryptosystems have a common property: they rely on a shared secret between communicating parties. This secret is used both as an encryption key and as a decryption key (thus the keyword "symmetric" in the name). This type of cryptography ensures only confidentiality and fails to provide the other three goals of cryptography. Even more importantly, the disadvantage of symmetric key cryptography is that it cannot handle large communication networks. If a node in a communication network of n nodes needs to communicate confidentially with all other nodes in the network, it needs $n - 1$ shared secrets. For large n this is highly impractical and inconvenient. On the other hand, the big advantage over public key cryptosystems is that symmetric cryptosystems require much smaller key sizes for same level of security. Therefore, computations are much faster, the memory requirements are much smaller, and much smaller computational power is sufficient.

2.3 Public Key Cryptosystems

In public key cryptography there are two different keys, one called a *public key* is the key that should be publicly known, while the so-called *secret key* should be kept secret by the owner. The birth of the public key cryptography is credited to Whitfield Diffie and Martin Hellman. The powerful idea that they introduced was to use a mathematically intractable and computationally infeasible problem(s) as a security basis. The system is asymmetric since there are two different keys used: public key and private key. If data is encrypted with public key, it can only be decrypted using the corresponding private key, and vice versa. Today, all public key cryptosystems rely on some intractable problem. For example, cryptosystem RSA relies on difficulty of factoring large integers, while El-Gammal cryptosystem relies on the discrete logarithm problem (DLP) which is the problem of finding a logarithm of a group element with generator base in finite abelian group. Originally proposed Diffie-Hellman protocol relies on DLP as well. The beauty is that all public key cryptosystems do not need to have a shared secret between communicating parties. This solves the problem of large confidential communication network introduced earlier. In addition, public key cryptography opened door for ways of implementing technologies to ensure all four goals of cryptography. By means of combining the public key cryptography, public key certification, and secure hash functions, there are protocols that enable digital signatures, authentication, data integrity and non-repudation. Unfortunately, due to processor speed growth and even more due to smart modern cryptanalysis, the key sizes for public key cryptography grew very large. This gave a disadvantage in comparison to symmetric key cryptosystems: public key cryptography is significantly slower, requires large memory capacity, and large computational power. Just for comparison 128bit key used with DES cryptosystem has approximately the same level of security as the 1024bit key used with RSA cryptosystem [AMV97]. To solve these problems, researchers introduced different approaches. In order to decrease key sizes so the public key cryptography can be used in smaller computational environments (such as smart cards or handheld wireless devices), Neil Koblitz introduced the idea of using more exotic group in the public key underlying algebraic structure: the elliptic curve group. Elliptic curve cryptography (much of whose implementation is credited to CertiCom) enables smaller key sizes of public key cryptosystems that rely on DLP. The elliptic curve group algebra is much more complex so the cryptanalysis is much harder,

resulting in smaller key requirements. Another solution came from public key cryptosystems that initially use more complex computational problem, such as lattice reduction problem. Relatively new cryptosystem NTRU [JH98], based on the algebra of a ring of truncated polynomials, relies on lattice reduction problem and it is the only public key cryptosystems that have the speed, memory, and computational complexity comparable to symmetric key cryptosystems. However, since this system is relatively new, the underlying security of NTRU is yet to be investigated.

There is more popular approach in resolving the complexity of public key cryptosystems and that is combining the existing symmetric key cryptography with existing public key cryptography. This hybrid approach is very much widely accepted since it enables the optimal choices of the hybrid components. A fast symmetric key cryptography can be used in most of the lengthy communication, but the required shared secret can be shared on the fly using some public key scheme. Furthermore, the same public key scheme can be used to accomplish other three goals of cryptography. The most important is to enable speedy performance on the lengthy communication material. Using public key cryptography only to transmit the symmetric key or to authenticate digital signature will not significantly affect the performance.

2.4 Communication Cryptography vs. Digital Rights Management Cryptography

We must be careful to distinguish between the classical scenario of communication security and the relatively new scenario of security against piracy. More often than not, the modern theoretical cryptography addresses only the communication security. However, the industrial demand for digital rights management (DRM) is growing larger every day. Now, when multimedia capability of cell phones and small handheld wireless devices is getting closer and closer to the capability of multimedia desktop machines, industry is more and more interested in cryptographic approaches that offer solutions to protection against piracy.

Communication cryptography assumes that there are n trusted parties that share information and that cryptography is used in securing the communication channel. An adversary is undesired, non-trusted party

that wants to access the communication information by listening to the communication channel. This model assumes that the channel end-points are trusted.

In contrast, DRM cryptography deals with a scenario in which unicasting, multicasting, or broadcasting party needs to securely transmit the content to the pool of trusted devices. In this model, an adversary is not only the channel eavesdropper. The users of the trusted device can also potentially be adversaries. A user of the trusted device can view the content but should not be able to distribute the content. Eavesdropper shouldn't be able to neither view nor distribute the valuable content.

3 Multimedia Content Cryptography

In this section, we present the relevant topics regarding cryptography in respect to multimedia data.

3.1 Multimedia and Multimedia Security

Multimedia is a combination of the following media: text, still images, audio data, animation, and video. Multimedia security in general is a method or a set of methods used to protect the multimedia content. These methods are heavily based on cryptography and they enable either communication security, or security against piracy (DRM security), or both.

Communication security of digital images and textual digital media can be accomplished by means of standard symmetric key cryptography. Such media can be treated as binary sequence and the whole data can be encrypted using cryptosystem such as AES or DES [Sti02]. In general, when the multimedia data is static (not a real-time streaming) we can treat it as a regular binary data and use the conventional encryption techniques. Encrypting the entire multimedia stream using standard encryption methods is referred to as the *naive algorithm* [AG96].

However, due to variety of constraints (such as near real-time speed), communication security for streaming audio and video media is harder to accomplish. Communication encryption of video and audio multimedia

content is not simply the application of established encryption algorithms, such as DES or AES, to its binary sequence [JWJ01]. It involves careful analysis to determine and identify the optimal encryption method when dealing with audio and video media. Current research is focused on modifying and optimizing the existing cryptosystems for real-time audio/video. It is also oriented towards exploiting the format specific properties of many standard video and audio formats, in order to save desired speed and enable real-time streaming. This is referred to as *selective encryption* (see section 5.3) [JWJ01]. Selective encryption is particularly applicable to digital cable videos. For textual media and some low-quality audio and video streaming multimedia we can still apply real-time packet encryption by means of SRTP (Secure Real-time Transport Protocol) [RBO01], which is based on AES and encrypts entire multimedia bitstream.

3.2 Identifying Encryption Level in Multimedia Communication Security

There are many different occasions where multimedia communication security is desired. However, deciding upon what level of security is needed is harder than it looks. To identify an optimal security level, we have to carefully compare the cost of the multimedia information to be protected and the cost of the protection itself. If the multimedia to be protected is not that valuable in the first place, it is sufficient to choose relatively light level of encryption. On the other hand, if the multimedia content is highly valuable or represents a government or military secrets, the cryptographic security level must be the highest possible.

For many real-world applications such as pay-per-view, the content data rate should be very high, but the monetary value of the content may not be high at all. Thus, very expensive attacks are not attractive to adversaries, and light encryption may be sufficient for distributing such MPEG videos. For these applications, DRM is of much more interest.

On the other hand, applications such as videoconferencing or videophone (or even Internet phone) may require much higher level of confidentiality. If the videoconference is discussing important industrial, governmental or military secrets, the cryptographic strength must be substantial. Maintaining such high level of security and still keeping a real-time and limited-bandwidth

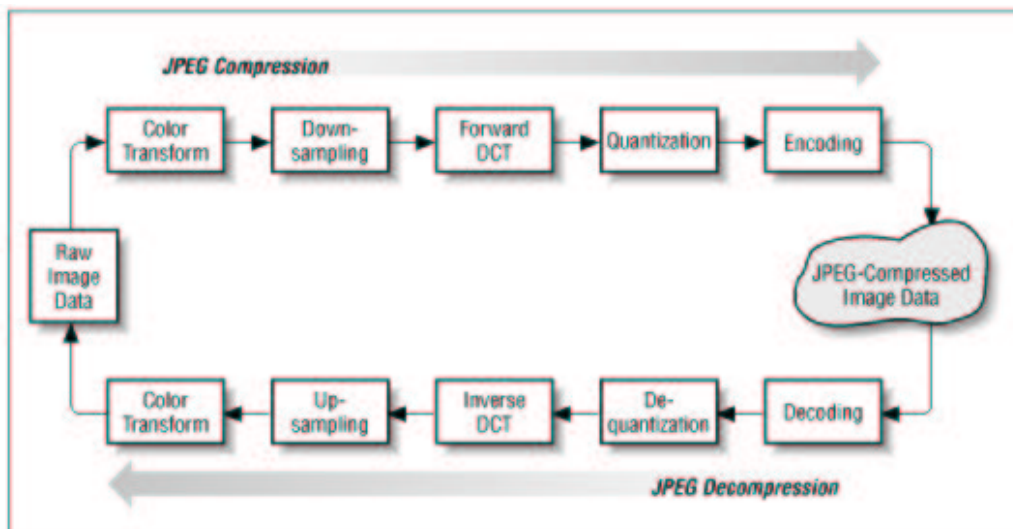
constraints is not easy to accomplish.

4 Overview of Main Multimedia Formats: JPEG and MPEG

To really understand the modern approaches to the multimedia security, it is necessary to understand the structure of the JPEG and MPEG formats. Due to the popularity, most of the proposed schemes are aimed to JPEG and MPEG formats, however, some basic ideas can be naturally extended to other formats since there are similarities between them. For example similar techniques and algorithms for MPEG format can be extended to H.26x family.

4.1 JPEG

JPEG (Joint Photographic Expert Group) format is standardized in 1992. This is a lossy type of compression algorithm whose quality and compression ratio are inversely proportional (tradeoff). The figure below shows the stages of JPEG encoding and decoding:



Stages in the JPEG compression algorithm

In short, the image is divided into 8x8 blocks which undergo several stages, such as DCT (Discrete Cosine Transform), quantization (lossy stage), zig-zag sequencing, and entropy encoding based on Huffman compression. The DCT transformed 8x8 block has the property that most information carrying bytes are the ones located at the top left corner of the block. JPEG images are building blocks for MPEG videos, which are discussed next.

4.2 MPEG

The MPEG acronym stands for Moving Picture Experts Group and it is the name of the one of the most important and most widely used video codecs. The MPEG1-2 video encoding scheme represents the video signal using the repetition of group of pictures (GOPs). Each GOP is a sequence of selected I, P and B frames. Examples of GOP sequences are (IBBBPBBB) or (IBBPBBPBB), but the relative frequency of I, P and G frames may be application dependent. I frames are encoded as standard JPEG images, without reference to other frames. Consequently, I frames are of the smallest compression ratio. On the other hand, P frames are encoded with reference to the I frames, containing only the differences between the two consecutive frames. Since the time difference between two consecutive frames is a fraction of a second, the Hamming distance between pixel blocks is very small. Therefore, P frames have much better average compression ratio than the I frames. Finally, B frames are bidirectionally interpolated using the previous closest I/P frame and the following closest I/P frame. The average compression ratio of B frames is usually the highest.

5 Communication Encryption of Video and Audio Multimedia

In this section we discuss different research directions which were taken regarding communication encryption of multimedia.

5.1 Naive Approach

As discussed earlier, a naive approach can be used to encrypt the multimedia content. The whole multimedia stream is encrypted using some symmetric key cryptosystem. However, the real-time video and audio playback is not met by using modern symmetric schemes such as DES or AES, as discussed in [JWJ01], [AG96], [SWB99], and [QN98]. The next section discusses scrambling, a popular method of applying fast, yet very insecure encryption to the whole stream. Scrambling was the product of immediate industrial need by cable companies for a fast solution to make free viewing of paid cable channels more difficult than doing nothing to the signal. On the other hand, applying advanced encryption scheme (AES for example) on the bit stream ensures extremely good privacy but doing so for audio and video media is not a good idea.

5.2 Scrambling: Simple Encryption Attempt

Scrambling is one of the simplest form of encryption that can be applied to multimedia data. Even though there is no distinct definition of scrambling, it really refers to a simplest possible encryption attempts such as simple substitution or simple transposition ciphers whose security in today's world is lowest possible. Early work on signal scrambling was based on using an analog device to permute the signal in the time domain or distort the signal in the frequency domain by applying filter banks or frequency converters [CPW01]. However, these schemes are extremely easy to crack using modern computers. With the popularization of DSP (Digital Signal Processing), in the digital signal domain focus was placed on scrambling in the domain of orthogonal transforms (DFT, DCT, wavelet transform, Hadamard transform, etc.) [CPW01]. Unfortunately, these techniques, although more secure than the old analog ones, are still much less secure than the conventional encryption. In fact, they do not fulfill any of the goals of cryptography. As far as the attacker (adversary), those methods represent a temporary inconvenience in getting the original video or audio signal.

There is another problem associated with scrambling. Scrambling of raw signal decreases the effectiveness of the compression algorithms [CPW01]. Compression algorithms have been designed for the unscrambled signals and they use the statistical characteristics of raw data. Once the signal

is scrambled, these characteristics will change and the performance of the compression filter will be degraded. Thus, encryption schemes that are completely compatible with multimedia compression are more favorable.

5.3 Selective Encryption

In order to meet real-time constraint for audio and video multimedia playback, selective encryption is used. The basic idea of selective encryption is to encrypt only a portion of the compressed bitstream. For example, we can select only the most important coefficients from either final or intermediate steps of a compression process and encrypt those. Less important coefficients are left unencrypted or lightly scrambled. Typical examples of selective encryption for videos are Secure MPEG by Agi and Gong [AG96], zig-zag permutation by Tang [Tan96], VEA by Qiao and Nahrstedt [QN98], VEA1-4 by Bhargava et al. [SWB99], selective scrambling by Zeng and Lei [ZL02], and a format-compliant configurable encryption by Wen et al. [JWJ01], and for audio and speech perception-based selective encryption by Servetti and De Martin [AS02]. Next, we present and analyze some important solutions based on selective encryption from the recent research publications.

Focus 1: Secure MPEG (SECMPEG)

In [MG95] Meyer and Gadgetast introduced a selective encryption method applicable to MPEG standard. SECMPEG is designed with four different levels of security. At the first level, SECMPEG encrypts the headers from the sequence layer to the slice layer, while the MVs and DCT blocks are unencrypted. At the second level, most relevant parts of the I-blocks are encrypted (upper left corner of the block) in addition. At the third level, SECMPEG encrypts all I-frames and all I-blocks. Finally, at the fourth level, SECMPEG encrypts the whole MPEG sequence (naive approach).

The selective encryption in SECMPEG (levels 1, 2, and 3) has some weaknesses [AG96], [QN98]. It is shown that even though single P- or B-frame on its own carries almost no information without the corresponding I-frame, a series of P- or B-frames can tell a lot if their base I-frames are correlated. The experiments in [AG96] by Agi and Gong show that encrypting I-frames only may leave visible I-blocks in other frames. The same authors then propose a few trade-off improvements: increasing the frequency of the I-frames and/or

encrypting all P- and B-frames. These improvements decrease speed, and further disrupt the compression ratio.

Since SECMPPEG introduces changes to the MPEG format, a special encoder and decoder is needed to handle SECMPPEG streams.

Focus 2: Zig-Zag Permutation Algorithm by Tang

Zig-zag permutation algorithm have encryption embedded into the MPEG compression process. As we saw in earlier discussion, JPEG and I-frames of MPEG undergo a zig-zag reordering of the 8x8 blocks. The zig-zag pattern forms a sequence of 64 entries that are ready to enter entropy encoding stage (compression). The main idea of Tang's approach is to use a random permutation list to map the individual 8x8 blocks to a 1x64 vector.

The Algorithm is consisted of three stages:

- STAGE 1: A list of 64 permutations is generated.
- STAGE 2: Splitting procedure on an 8x8 block is performed as follows: We denote DC coefficient by an 8 digit binary number with digits $b_7b_6b_5b_4b_3b_2b_1b_0$. This binary sequence is then split into two equal halves: $b_7b_6b_5b_4$ and $b_3b_2b_1b_0$. Finally, we place number $b_7b_6b_5b_4$ into DC coefficient and the number $b_3b_2b_1b_0$ as the AC63 (the last AC coefficient) which is the least important value in the block. This will result in no visible degradation of the quality.
- STAGE 3: The random permutation is applied to the split block.

However, the zig-zag permutation algorithm is not particularly secure [QN98]. Qiao and Nahrstedt introduced two types of attacks on zig-zag permutation: known-plaintext attack, and ciphertext-only attack. Known-plaintext attack is particularly applicable to videos with known clips such as blank screens, the movie rating screens, MGM roaring lion, etc. If these known frames are used as a comparison, the adversary can easily generate the "secret" permutation list. Since Tang himself realized large vulnerability to known-plaintext attack, he introduced an improvement of the zig-zag permutation technique by using the *binary coin flipping sequence* method together with two different permutation lists. Two different

permutation lists are generated and for each 8x8 block a coin is randomly flipped. The outcome event determines which list to apply; i.e., if the head is flipped we apply one list and if the tail is flipped we apply the other list. As shown in [QN98], this addition turns out to be useless. If we know some of the original frames in advance (known-plaintext) then by simple comparison both permutation lists can be found. Since of the certain statistical properties of the blocks (upper left corner gathering of the AC coefficients within a block) we can easily determine which list of permutations is used. In addition, Qiao and Nahrstedt showed that the Tang's splitting method is just a subset of SECMPPEG with second level, and thus its security is not good.

Finally, Qiao and Nahrstedt showed that the Tang's zig-zag permutation algorithm is susceptible to the ciphertext only attack. The attack relies on the statistical properties of the DCT coefficients where most nonzero terms are gathered in the top left corner of the I-block. Statistical analysis shows the following observation [QN98]: DC coefficient always has the highest frequency value, the frequency of AC1 and AC2 coefficients is among top 6, and frequency of AC3 and AC4 is among top 10. Applying these will reconstruct the original video to the pretty good accuracy since first five DCT coefficients carry most information about the block.



I-frame decoded using only DC coefficients



I-frame decoded using DC, AC1, and AC2 coefficients



I-frame decoded using DC, AC1, AC2, AC3 and AC4 coefficients



The original I-frame

As shown in the figures above, cracking the first five DCT coefficients of the I-blocks would result in the good quality decryption using ciphertext only.

Therefore, zig-zag permutation cipher is a weak one.

Focus 3: VEA by Qiao and Nahrstedt

The Video Encryption Algorithm by Qiao and Nahtstedt [QN98] is constructed with goal to exploit the statistical properties of the MPEG standard. The algorithm consists of several steps:

- STEP 1: Let the $2n$ byte sequence denoted $a_1a_2 \dots a_{2n}$ represent the chunk of an I-frame.
- STEP 2: Create two lists, one with odd indexed bytes, and the other with even indexed bytes.
- STEP 3: Xor the two lists into an n byte sequence denoted with $c_1c_2 \dots c_n$.
- STEP 4: Apply the chosen symmetric cryptosystem E (for example DES or AES) on odd list and create ciphertext $c_1c_2 \dots c_nE(a_2a_4 \dots a_{2n})$.

The security of this method is very close to the security of the encryption scheme E that is internally used. The speed of this algorithm is roughly 1/2 of the speed of naive algorithm, but that is still large amount of computation for real-time high quality videos [CPW01].

Focus 4: VEAs by Bhargava, Shi, and Wang

In [SWB99], Bhargava, Shi, and Wang introduced four different video encryption algorithms: algorithm I, algorithm II (VEA), algorithm III (MVEA) and algorithm IV (RVEA).

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 π which is used to permute standard JPEG/MPEG Huffman codeword list. In order to save compression ratio, the permutation π must be such that it only permutes the codewords with same number of bits. There is yet another limitation: the distance between the original and permuted codeword set must be greater than the encryption quality parameter β .

The security of algorithm I is not particularly good. In [SS03], Socek and Sramka showed that algorithm I is highly vulnerable to both known-plaintext attack, and ciphertext-only attack. If some of the video frames are known in advance (such as standard introductory jingles and similar), one can reconstruct the secret permutation π by comparing the original and encrypted frames. Vulnerability to this type of attack was also discussed in [SWB99]. However, algorithm I is also subject to ciphertext-only attack. Socek and Sramka defined the *low-frequency error attack* on algorithm I ciphertext. Basically, since permutation π 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 number 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 π shuffle the 16bit codewords. But these are appearing extremely rare, and the reconstructed video may be of almost the same quality as the original.

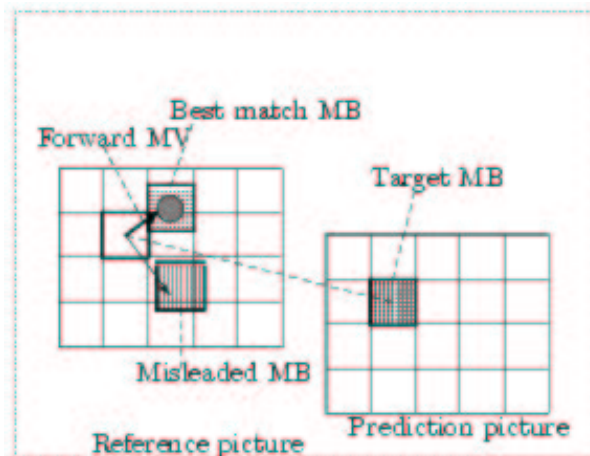
Algorithm II (VEA) uses the following selective encryption: it is sufficient to encrypt only the sign bits of the DC coefficients in the I-frame blocks since they carry most of the information. VEA simply xors the sign bits of the DC coefficients with a secret binary key k .

The security of VEA really depends on the length of the key. The authors encourage a use of a long binary key. However, too long of a key may be infeasible and impractical. On the other hand if the key is short, the system can be easily broken. If the key is as long as the video stream and it is unique and used only once that would correspond to Vernam cipher which is absolutely secure. However, this is highly impractical for mass applications such as VOD (Video On Demand) and similar. On the other hand if the key is too short, the whole method becomes simply Vigenere cipher [SS03] which is a classical cipher for which there are many attacks developed (for example Kasiski analysis). In addition, Vigenere is highly sustainable to the known-plaintext attack (one can literally read off the secret key). Authors in [SWB99] suggest the use of the pseudo-random generator that generates a stream of pseudo-random bits to be the key k of arbitrary length. The trouble with this scheme is how synchronize P-RNG (Pseudo-Random

Number Generator), or how to securely transmit the random seed.

Algorithm III (MVEA) is an improvement to the VEA algorithm that includes the following additions: the sign bits of differential values of motion vectors in P- and B-frames can also be randomly changed. This type of improvement makes the video playback more random and more unviewable. When the sign bits of differential values of motion vectors is changed, the directions of motion vectors change, making the whole video very chaotic.

Unfortunately, the security issues and problems that are applicable to VEA are also applicable to MVEA.



Misleading motion generated by MVEA and RVEA

Algorithm IV (RVEA) is a more secure approach to MVEA algorithm. This approach is robust under both ciphertext-only attack and known-plaintext attack. The difference between RVEA and MVEA/VEA algorithms is that MVEA uses conventional symmetric key cryptography (e.g. DES, AES, or IDEA) to encrypt sign bits of DC coefficients and the sign bits of motion vectors. This selective approach significantly speeds up the process by only encrypting certain sign bits. The experiments show that the encryption quality is quite good considering the amount of information changed.

The following figures illustrate all the video encryption algorithms VEA, MVEA, and RVEA in action:



Original I-frame



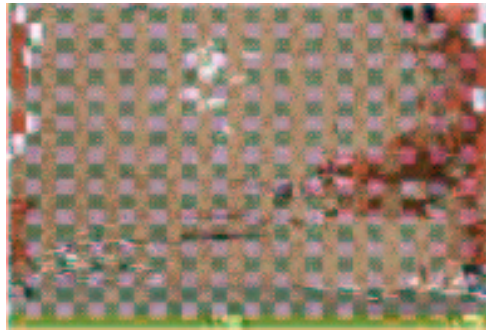
Encrypted AC coefficients with VEA



Encrypted ACs, DCs, Cr and Cb blocks with VEA



Encrypted all ACs and DCs with VEA



Encrypted motion vectors only with MVEA

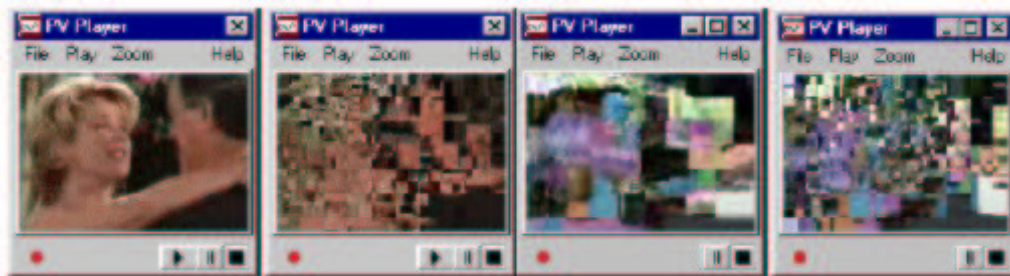


A RVEA encrypted I-frame

Focus 5: Format-compliant configurable encryption by Wen et al.

This approach, introduced in [JWJ01] by Wen et al., is generalizing the ideas of selective encryption into format-compliant method. It stresses

the importance of maintaining the compliance to a standard (such as MPEG codec). In this model, the data is divided into information-carrying and not information-carrying portions. We only need to encrypt the information-carrying fields. These fields are either fixed length code (FLC) codewords, or variable length code (VLC) codewords. To achieve the format compliance, we extract the bits that are chosen to be encrypted, concatenate them, and then encrypt with secure scheme such as DES. Finally, we put the encrypted bits back to their original positions. For FLC coded fields, this operation most likely creates compliant bitstream. If it doesn't, we apply same approach as for VLC coded fields. For VLC coded fields, encrypting concatenated codewords may not result in another valid codeword. In order to still maintain compliance, we assign a fixed length index to each codeword in VLC codeword list, encrypt the index concatenation and map it back to the codewords. This however introduces the overhead. Percentage of overhead is content-dependent.



Example of format-compliant approach (left to right): The original, Encrypted VLC only, Encrypted FLC only, and Encrypted VLC and FLC

5.4 New Multimedia-Specific Encryption Algorithms

There are few notable special encryption algorithms that were developed specifically for audio and video data. Most of them rely on parallel hardware implementations and not on the software solutions. For example, Li et al. introduced *chaotic video encryption scheme* (CVES) in [LZMC02]. This scheme is based on multiple digital chaotic systems. Mostly, an efficient hardware implementation is discussed. However, there are some security concerns discussed by Li and Zheng in [LZ02]. Li and Zheng showed that

the original claim about the resistance level to a ciphertext-only attack was overestimated. In addition, it shows that the system is easily cracked under known- and chosen-plaintext attacks. In fact, if only one plain image and its cipher image is known, system is broken.

6 DRM for video and audio multimedia

We have discussed only one type of encryption, namely the encryption used to protect the communication channel. The second type of encryption is often referred to as Digital Rights Management, and this type of encryption has been given real attention only recently. The ideas for DRM include a number of solutions and approaches which are worth writing survey paper on its own. This type of cryptography is in a way harder to achieve since there are much more implementational issues, such as tamper resistant devices, key management, authentication, TA (Trusted Authority), key certification, format compliance, etc. Many systems were created and then they failed since not all of these issues were properly addressed. Examples of these are the DVD protection format CSS, and the electronic book protection format, both of which were successfully cracked. Current efforts are switching to more friendly neighborhood of watermarking. Digital watermarks cannot prevent the adversary to steal the media content, but it can be proven that he or she is not a legal owner of the content, and the adversary can be prosecuted. Fear of prosecution should decrease if not stop illegal distribution of watermarked media.

7 Conclusions and Further Research

We haven't seen a solid solution to communication security of the audio and video data. Even more so, we haven't seen successful DRM technology for media content. The future work should be directed towards improving existing propositions so that in the near future there are standardized solutions available for different multimedia security needs.

References

- [AG96] I. Agi and L. Gong. An empirical study of secure mpeg video transmission. *In Symposium on Network and Distributed Systems Security. IEEE*, 1996.
- [AMV97] P.C. van Oorschot A.J. Menezes and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., 1997.
- [AS02] J.C. De Martin A. Servetti. *Perception Based Partial Encryption of Compressed Speech*. IEEE Transaction on Speech and Audio Processing, Vol. I, No. 8, 2002.
- [CPW01] C.-C. Jay Kuo C.-P. Wu. Efficient multimedia encryption via entropy codec design. *SPIE International Symposium on Electronic Imaging 2001, San Jose, CA, USA*, 4314, 2001.
- [JH98] J.H. Silverman J. Hofhstein, J. Pipher. *NTRU: A Ring-Based Public Key Cryptosystem*. Preprint, 1998.
- [JWJ01] W. Zheng M. Luttrell J. Wen, M. Severa and W. Jin. *A Format-Compliant Configurable Encryption Framework for Access Control of Multimedia*. Proceedings International Workshop on Multimedia Signal Processing, pp. 435-440, 2001.
- [LZ02] Shujun Li and Xuan Zheng. Cryptanalysis of a chaotic image encryption method. *In Proceedings of 2002 IEEE International Symposium on Circuits and Systems (ISCAS 2002), vol. II*, pages 708–711, 2002.
- [LZMC02] Shujun Li, Xuan Zheng, Xuanqin Mou, and Yuanlong Cai. Chaotic encryption scheme for real-time digital video. *In Real-Time Imaging VI*, Proceedings of SPIE vol. 4666, pages 149–160, 2002.
- [MG95] J. Meyer and F. Gadegast. Security mechanisms for multimedia-data with the example mpeg-1-video. *Proj. description of SEC MPEG, Tech. Univ. of Berlin, Germany*, 1995.

- [QN98] Lintian Qiao and Klara Nahrstedt. Comparison of MPEG encryption algorithms. *Computers and Graphics*, 22(4):437–448, 1998.
- [RBO01] D. McGrew M. Naslund K. Norman R. Blom, E. Carrara and D. Oran. *The Secure Real time Transport Protocol (SRTP)*. Internet draft, 2001.
- [SS03] D. Socek and M. Sramka. Cryptanalysis of video encryption algorithms. *in preparation*, 2003.
- [Sti02] D.R. Stinson. *Cryptography Theory and Practice*. CRC Press, Inc., 2002.
- [SWB99] C. Shi, S. Wang, and B. Bhargava. Mpeg video encryption in real-time using secret key cryptography. *in Proc. of PDPTA '99, (Las Vegas, Nevada)*, 1999.
- [Tan96] Lei Tang. Methods for encrypting and decrypting MPEG video data efficiently. In *ACM Multimedia*, pages 219–229, 1996.
- [ZL02] Wenjun Zeng and Shawmin Lei. Efficient frequency domain selective scrambling of digital video. *IEEE Transactions in Multimedia*, 2002.