Tatra Mt. Math. Publ. **29** (2004), 79–87

TATRA MOUNTAINS
Mathematical Publications

CRYPTANALYSIS OF VIDEO ENCRYPTION ALGORITHMS

Tanya E. Seidel — Daniel Socek — Michal Šrámka

ABSTRACT. Content security is an important issue in multimedia applications. In this paper we perform a cryptanalysis of several encryption algorithms that have been proposed to protect the privacy of MPEG video streams. In particular, we analyze the encryption based on permuting the Huffman codeword list, and two selective encryption algorithms—VEA and MVEA. For the convenience of the reader, we provide a simple overview of MPEG encoding, and we include all analyzed algorithms and their characteristics.

1. Introduction

Multimedia content is a combination of any of the following media: text, still images, audio, animation, and video. Multimedia security deals with ways of protecting such content. In general, this is achieved by methods that are heavily based on cryptography. These methods enable communication security, piracy protection (so called Digital Rights Management), or both.

Communication security of multimedia content can be accomplished by means of standard symmetric-key cryptography. In particular, viewing multimedia content as a sequence of binary data, protection can be thought of as applying conventional symmetric-key encryption techniques (such as AES) to the whole sequence. This method is referred to as a *naive algorithm*. Unfortunately, due to a variety of constraints, applying the naive approach to more complex multimedia streams (mostly video and audio) creates significant computational overhead.

Communication encryption of video and audio content is therefore harder to accomplish. It involves a careful analysis to determine and identify the optimal encryption method when dealing with audio and video content. Current research is focused on modifying and optimizing the existing cryptosystems for real-time audio and video content. It is also oriented towards exploiting the

2000 Mathematics Subject Classification: 68P25, 94A60.
Keywords: cryptanalysis, symmetric ciphers, MPEG encryption.

specific properties of many standard video and audio formats, in order to achieve desired speed and enable real-time streaming. This is referred to as *selective encryption* [4].

The challenges of video encryption come from several facts. First, the size of a typical MPEG compressed video file is often very large (for example, the size of a two hour MPEG-1 video is about 1 GB). Second, the decoding (as well as decryption) needs to be processed in real time (for example, the MPEG-2 video streams can be as large as 40 Mbps). Third, VCR-like functions such as fast forward or playback from any point should be available and reasonably fast. In addition, there is more to video security than just encryption. The already mentioned Digital Rights Management is an example of a security requirement that goes beyond communication security.

For the convenience of the reader, a simple introduction to MPEG-1 video encoding, necessary for understanding the encryption algorithms, is provided. The cryptanalysis of some of the proposed MPEG-1 video encryption algorithms follows. In particular, we present practical attacks on two selective MPEG-1 video encryption algorithms—VEA and MVEA, and one light-weight MPEG-1 video encryption algorithm based on permuting the Huffman codeword list.

2. MPEG-1 video encoding

While we do not present a detailed description of the MPEG-1 video encoding, the following paragraphs outline the core components of this process.

The MPEG-1 video encoding [2] 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. Typical GOP sequences are IBBBPBBB or IBBPBBPBB, but the relative frequency of I, P and B frames may be application dependent. I frames are encoded as standard JPEGs, without reference to other frames. Consequently, I frames are of the smallest compression ratio. A P frame is encoded with reference to the previous I frame, containing only the difference between these two frames. Since the time difference between these two frames is a fraction of a second, the difference between blocks of pixels is very small. Therefore, P frames have a 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.

The encoding of I frames differs from that of the P and B frames. An I frame is encoded as a standard JPEG still image. JPEG encoding is a complex process. The JPEG encoding [3] is a lossy type of compression, a tradeoff between quality of the image and compression ratio. For our purposes, it is sufficient to know that an image is first divided into blocks of 8×8 pixels. All JPEG encoding operations are then performed on these blocks. The encoding starts with a Discrete Cosine Transformation (DCT), which computes the DC coefficient—the coefficient containing the crucial information of the whole block, and 63 AC coefficients. The next stages are quantization, the lossy stage, and zig-zag sequencing. Finally, the block undergoes entropy encoding using the Huffman code, where the codeword list is fixed.

3. Cryptanalysis of VEA and MVEA

VEA and MVEA are MPEG-1 video encryption algorithms that were introduced in [1]. We provide a description of the algorithms before stating our cryptanalytic results.

Table 1 and Table 2 present the simplified pseudocode of the VEA and MVEA algorithms, respectively. From the pseudocode, it is clear that VEA and MVEA are almost the same. The only difference is the set of video stream bits that are encrypted. In both cases, the algorithm works with macro blocks of size 16×16 pixels subsampled into four 8×8 blocks Y representing luminescence, and two chrominance 8×8 blocks, Cr and Cb. For a more detailed description, refer to [1].

Table 1. VEA encryption.

Input: key k of length m bits, video bit-stream v of length n

Output: encrypted video bit-stream w

Algorithm:

for every bit $i = 1, \ldots, n$ in video bit-stream v do

1. if v_i is a beginning of GOP, then $j \leftarrow 0$
2. else if v_i is a sign bit of DC coefficient or a sign bit of DC differential value of Y, Cr, or Cb block of I frame, then
   (a) $w_i \leftarrow v_i \oplus k_j$
   (b) $j \leftarrow j + 1 \pmod m$
3. else $w_i \leftarrow v_i$

As mentioned before, there is no need to encrypt the video bit by bit. The VEA and MVEA algorithms take advantage of this concept. That is, they perform encryption on predetermined (fixed) bits of the video stream, and therefore fall into the category of selective encryption algorithms.

One of the most significant properties of VEA and MVEA from [1] is the following: One can encrypt a MPEG-1 video many times, and decrypt it in one step. Let $E_k(P)$ denote the VEA or MVEA encryption of plaintext P using key k. For any two distinct keys k_1 and k_2, it holds that

$$E_{k_1}(E_{k_2}(P)) = E_{k_1 \oplus k_2}(P). \tag{1}$$

Therefore the decryption can be performed in one step using the key $k_3 = k_1 \oplus k_2$.

This fact is very useful in the case when a key needs to be changed. The change is cost-effective since the decryption process can be eliminated, unlike in other cases, where decryption must be performed before obtaining new ciphertext. In our analysis of VEA and MVEA, we will show that this key-change has no impact on the security.

TABLE 2. MVEA encryption.

Input: key k of length m bits, video bit-stream v of length n

Output: encrypted video bit-stream w

Algorithm:

for every bit $i = 1, \dots, n$ in video bit-stream v do

1. if v_i is a beginning of GOP, then $j \leftarrow 0$
2. else if v_i is a sign bit of DC differential value of Y, Cr, or Cb block of I frame, then
   (a) $w_i \leftarrow v_i \oplus k_j$
   (b) $j \leftarrow j + 1 \pmod{m}$
3. else v_i is a sign bit of differencial value of motion vectors of B or P frame, then block of I frame, then
   (a) $w_i \leftarrow v_i \oplus k_j$
   (b) $j \leftarrow j + 1 \pmod{m}$
4. else $w_i \leftarrow v_i$

3.1 Attacking VEA and MVEA

Let P denote the collection of compressed video bits for a single GOP that needs to be protected. P can be represented as $P = (p_0, p_1, \dots, p_{t-1})$, where, in the case of VEA, p_i for $i = 0, \dots, t-1$ are all the sign bits of the DC coefficients, and all the sign bits of the discrete cosine differential value of a Y, Cr, or Cb block of an I frame, in their original order. When considering MVEA, the p_i's are all the sign bits of the discrete cosine differential values of a Y, Cr, or Cb block of an I frame, and all the sign bits of the differential values of the motion vectors of B and P frames, in their original order. The authors of the original algorithms give an analytical explanation as to why encrypting only these bits of the video stream leads to sufficient protection.

Let k be a key of length m bits. Again, k can be represented as a finite bit-stream $k = (k_0, k_1, \ldots, k_{m-1})$, where $k_i \in \{0, 1\}$ for every $i = 0, \ldots, m-1$. The key k should be expanded to length t by repeatedly concatenating the bits $k_0, \ldots, k_{m-1}$ and taking the first t bits of such concatenation. The VEA's and MVEA's encryption function $E_k(P)$ for a single GOP is defined as

$$E_k(P) = (c_0, c_1, \ldots, c_{t-1}), \tag{2}$$

where $c_i = p_i \oplus k_{(i \bmod m)}$ for every $i = 0, \ldots, t-1$, or if we consider the expanded key k, each c_i is simply $c_i = p_i \oplus k_i$.

Consider the property in equation (1). It turns out that this property is quite useless. Suppose that the original key k was compromised or needed to be changed for some reason. As was stated before, decryption is not necessary to obtain a new ciphertext. Hence one only needs to encrypt the original ciphertext $E_k(P)$ with a new key, say ℓ, to obtain a new ciphertext. The new ciphertext can be then decrypted using the (expanded) key $k \oplus \ell$.

Now, take any plaintext bit p_i, for some $0 \le i < t$. The old and new ciphertexts are

$$E_k(p_i) = p_i \oplus k_i, \tag{3}$$

$$E_\ell\big(E_k(p_i)\big) = (p_i \oplus k_i) \oplus \ell_i, \tag{4}$$

respectively, where k and ℓ are considered to be expanded keys. Values of both equations represent a ciphertext, and so they are known to the attacker. By subtracting equation (3) from equation (4), one can obtain ℓ_i. This can be done for any i, and thus one can easily obtain the whole key ℓ. Therefore, there is no security advantage in applying an additional encryption to an already encrypted content.

We conclude our analysis of VEA and MVEA algorithms by observing that the encryption function, as defined in equation (2), is exactly the definition of one of the classical ciphers—the Vigenère cipher. It immediately follows that a known-plaintext attack and a ciphertext-only attack are possible. For the latter attack, one can easily obtain the frequencies of selected bits that need to be encrypted by examining other MPEG-1 video streams. According to the analytical explanation as to why encrypting only these bits leads to sufficient protection [1], we believe that such frequency distribution cannot be uniform. Therefore it is possible to launch a ciphertext-only attack. For explanation of the attacks on the Vigenère cipher, see [6]. Also note, that because of the existence of resynchronization points at the beginning of each GOP that forces the key to start from its first bit, the length of the key k cannot be arbitrarily large.

4. Cryptanalysis of "Huffman" cipher

The "Huffman" cipher [5], is a light-weight MPEG-1 video encryption algorithm which incorporates encryption and decryption with MPEG-1 video encoding and decoding, respectively, in one step. First, the algorithm description from [1] and properties of this algorithm are stated. Our cryptanalysis of the algorithm follows.

Let H be the Huffman codeword list provided by the JPEG-1 standard. Let H' be another permutation of H, and let $\pi\colon H \to H'$ be mapping with the following properties:

1. π is bijective,
2. $|w| = |\pi(w)|$ for $\forall w \in H$, where $|\cdot|$ denotes the length (number of bits) of a given codeword,
3. $\|H - H'\| > \beta$, with β a sufficiently large user-selectable encryption quality constant, and $\|H - H'\|$ is the distance between H and H', defined as

$$\|H - H'\| = \sqrt{\sum_{w \in H} \big(w - \pi(w)\big)^2}\,.$$

In other words, π is a permutation of a list of codewords H which preserves the length of codewords.

The MPEG-1 video encryption and decryption is then embedded in compression and decompression, respectively, as follows:

- Encryption: during the MPEG-1 video compression process, H is replaced with H'.
- Decryption: during the MPEG-1 video decompression, for every word $w \in H'$, $\pi^{-1}(w)$ is used as the real Huffman codeword value.

The characteristics of the algorithm are:

- No overhead is added—the encryption computation time is decreased by combining MPEG-1 compression and encryption (by replacing Huffman codeword list H with H'). The same applies to decompression and decryption. In other words, these processes do not result in extra computation time.

- The same compression ratio is achieved—because of the property that for every $w \in H$: $|w| = |\pi(w)|$, the compressed and encrypted output from the MPEG-1 encoding process will have exactly the same size as if encryption was not included.

4.1 Two different attacks

The first observation is that given encrypted video content and its corresponding original content, we can apply a known-plaintext attack. Thus, by examining I frames and their encrypted equivalents, it is not at all hard to determine the permutation π which serves as a key in this cipher.

A ciphertext-only attack involves exploiting the properties of the Huffman code. By the construction of the Huffman code, shorter codewords are assigned to symbols of more frequent input. Consequently, input symbols which rarely appear are assigned long codewords.

There are two fixed Huffman codeword lists used in standard JPEG encoding. The first codeword list, summarized in Table 3, is used to encode the DC coefficients, while the second list, summarized in Table 4, is used to encode the AC coefficients.

Table 3. Huffman codeword list for DC coefficients.

length of codewords	2	3	4	5	6	7	8	9
number of codewords	1	5	1	1	1	1	1	1

Table 4. Huffman codeword list for AC coefficients.

length of codewords	2	3	4	5	6	7	8	9	10	11	12	15	16
number of codewords	2	1	3	3	2	4	3	5	5	4	4	1	123

There are only $5! = 120$ possible permutations for the first Huffman codeword list. However, there are

$$2! \cdot 3! \cdot 3! \cdot 2! \cdot 4! \cdot 3! \cdot 5! \cdot 5! \cdot 4! \cdot 4! \cdot 123! \approx 2^{718}$$

possible permutations of the second list, leading to an extremely large keyspace. By ignoring permutations involving long codewords (length 16), that is, by ignoring rare AC coefficients, the keyspace of the second list can be reduced to

$$2! \cdot 3! \cdot 3! \cdot 2! \cdot 4! \cdot 3! \cdot 5! \cdot 5! \cdot 4! \cdot 4! \approx 2^{38}$$

allowing for a computationally feasible exhaustive search. In this case, we start by selecting the permutation π such that it fixes all the codewords of length 16.

Then there are roughly 2^{38} possibilities for completing the permutation such that $|w| = |\pi(w)|$ for every $w \in H$.

Experiments showed that this attack on a 512×512 pixel image (4096 blocks) resulted in decryption failure of only 4 blocks, roughly 0.1 % failure. All other blocks were decrypted successfully and allowed for almost no degradation of the original image. In addition, the incorrectly deciphered blocks can be approximated from their surrounding blocks. The reconstructed 512×512 pixel image, as produced by our attack, is in Figure 1.

FIGURE 1. Reconstructed JPEG's Lena picture.

5. Conclusion

Protecting multimedia content is a very important issue. On one hand, protection by a naive algorithm provides maximum security, but requires a special, costly hardware for real-time decryption. On the other hand, most of the multimedia applications need to balance between security and cost of video streaming. So there is an apparent tradeoff between security and the speed of streaming. The goal is to design a reasonably fast and secure encryption method such that breaking this method requires an investment several times higher than the value of the content.

Selective encryption is only one part of communication security. It is, however, a crucial part, and thus studying this kind of techniques is extremely important. We were able to cryptanalyze two selective MPEG-1 video encryption algorithms and one light-weight MPEG-1 video encryption algorithm based on permuting the Huffman codeword list. All three of these methods were designed for efficiency, but unfortunately, as we have illustrated, they all lack security.

REFERENCES

[1] BHARGAVA, B.—SHI, C.—WANG, S. Y.: *MPEG video encryption algorithms*, Multimedia Tools and Applications **24** (2004), 57–79.

[2] GALL, L. D.: *MPEG: A video compression standard for multimedia applications*, Communications of the ACM **34** (1991), 46–58.

[3] ITU-T: *T.81—Digital compression and coding of continuous-tone still images*, Switzerland, 1992.

[4] MEYER, J.—GADEGAST, F.: *Security mechanisms for multimedia-data with the example MPEG-1-video*, Proj. description of SECMPEG, Tech. Univ. of Berlin, Germany, 1995.

[5] SHI, C.—BHARGAVA, B.: in: Proc. of the Int'l Conf. on Multimedia (Multimedia 98, Shaping the Future), New Delhi, India, 1998, pp. 55–61.

[6] STINSON, D. R.: *Cryptography—Theory and Practice*, Second Edition, CRC Press, Singapore, 2002.

Received October 19, 2003

Tanya E. Seidel, Michal Šrámka
Department of Mathematical Sciences
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431
U. S. A.

E-mail: teseidel@aol.com
sramka@math.fau.edu

Daniel Socek
Department of Computer Science and Engineering
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431
U. S. A.

E-mail: dsocek@brain.math.fau.edu