

# On the Design of Perceptual MPEG-Video Encryption Algorithms

Shujun Li, Guanrong Chen, *Fellow, IEEE*, Albert Cheung, *Member, IEEE*, Bharat Bhargava, *Fellow, IEEE*, and Kwok-Tung Lo, *Member, IEEE*

**Abstract**—In this paper, some existing perceptual encryption algorithms of MPEG videos are reviewed and some problems, especially security defects of two recently proposed MPEG-video perceptual encryption schemes, are pointed out. Then, a simpler and more effective design is suggested, which selectively encrypts fixed-length codewords in MPEG-video bit streams under the control of three perceptibility factors. The proposed design is actually an encryption configuration that can work with any stream cipher or block cipher. Compared with the previously-proposed schemes, the new design provides more useful features, such as strict size-preservation, on-the-fly encryption and multiple perceptibility, which make it possible to support more applications with different requirements. In addition, four different measures are suggested to provide better security against known/chosen-plaintext attacks.

**Index Terms**—Cryptanalysis, fixed-length codeword (FLC), known/chosen-plaintext attack, MPEG, perceptual encryption.

## I. INTRODUCTION

THE WIDE USE of digital images and videos in various applications brings serious attention to the security and privacy issues today. Many different encryption algorithms have been proposed in recent years as possible solutions to the protection of digital images and videos, among which MPEG videos attract most attention due to its prominent prevalence in consumer electronic markets [1]–[5].

In many applications, such as pay-per-view videos, pay-TV and video on demand (VoD), the following feature called “perceptual encryption” is useful. This feature requires that the quality of aural/visual data is only *partially* degraded by encryption, i.e., the encrypted multimedia data are still partially perceptible after encryption. Such perceptibility makes it possible for potential users to listen/view low-quality versions

Manuscript received December 2, 2004; revised February 21, 2006. This research was supported in part by the City University of Hong Kong SRG under Grant 7001702, in part by The Hong Kong Polytechnic University’s Postdoctoral Fellowships Scheme under Grant G-YX63, in part by the Research Grant Council of Hong Kong under Grant PolyU 5232/06E, and in part by the National Science Foundation Grants ANI-0219110 and RIS-0292890.

S. Li and K.-T. Lo are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: hooklee75@hotmail.com; enktlo@polyu.edu.hk).

G. Chen is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: gchen@ee.cityu.edu.hk).

A. Cheung is with the Department of Building and Construction and Shenzhen Applied R&D Centres, City University of Hong Kong, Hong Kong (e-mail: clacc@cityu.edu.hk).

B. Bhargava is with the Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-2066 USA (e-mail: bb@cs.purdue.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2006.8888840

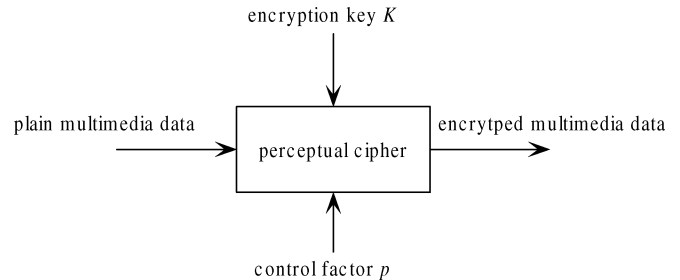


Fig. 1. Diagrammatic view of the perceptual encryption.

of the multimedia products before buying them. It is desirable that the aural/visual quality degradation can be continuously controlled by a factor  $p$ , which generally denotes a percentage corresponding to the encryption strength. Fig. 1 shows a diagrammatic view of perceptual encryption. The encryption key is kept secret (not needed when public-key ciphers are used) but the control factor  $p$  can be published.

Regarding the visual quality degradation of the encrypted videos, the following points should be remarked. 1) Since there does not exist a well-accepted objective measure of visual quality of digital images and videos, the control factor is generally chosen to represent a rough measure of the degradation. 2) The visual quality degradations of different frames may be different, so the control factor works only in an average sense for all videos. 3) The control factor is generally selected to facilitate the implementation of the encryption scheme, which may not have a linear relationship with the visual quality degradation (but a larger value always means a stronger degradation). 4) when the control factor  $p = 1$ , the strongest visual quality degradation of the specific algorithm (i.e., of the target application) is reached, but it may not be the strongest degradation that all algorithms can produce (i.e., all visual information of the video is completely concealed).

In recent years, some perceptual encryption schemes have been proposed for G.729 speech [6], [7], MP3 music [8], JPEG images [9], [10], wavelet-compressed (such as JPEG2000) images and videos [11]–[13] and MPEG videos [14]–[17], respectively. The selective encryption algorithms proposed in [18]–[20] can be considered as special cases of the perceptual encryption for images compressed with wavelet packet decomposition. In some research papers, a different term, “transparent encryption,” is used instead of “perceptual encryption” [16], [17], emphasizing the fact that the encrypted multimedia data are *transparent* to all standard-compliant decoders. However, transparency is actually an equivalent of another feature called “format-compliance” (or “syntax-awareness”) [21], [22],

which does not mean that some partial perceptible information in plaintexts still remains in ciphertexts. In other words, a perceptual cipher must be a transparent cipher, but a transparent cipher may not be a perceptual cipher [5]. Generally, perceptual encryption is realized by selective encryption algorithms with the format-compliant feature. This paper chooses to use the name of “perceptual encryption” for such a useful feature of multimedia encryption algorithms. More precisely, this paper focuses on the perceptual encryption of MPEG videos. After identifying some problems of the existing perceptual encryption schemes, a more effective design of perceptual MPEG-video encryption will be proposed.

The rest of this paper is organized as follows. The next section will provide a brief survey of related work and point out some problems, especially problems existing in two recently-proposed perceptual encryption algorithms [16], [17]. In Section III, the video encryption algorithm (VEA) proposed in [23] is generalized to realize a new perceptual encryption design for MPEG videos, called the perceptual VEA (PVEA). Experimental study is presented in Section IV, to show the encryption performance of PVEA. The last section presents the conclusion.

## II. RELATED WORK AND EXISTING PROBLEMS

### A. Scalability-Based Perceptual Encryption

Owing to the scalability provided in MPEG-2/4 standards [24], [25], it is natural to realize perceptual encryption by encrypting the enhancement layer(s) of an MPEG video (but leaving the base layer unencrypted) [14]. However, since not all MPEG videos are encoded with multiple layers, this scheme is quite limited in practice. More general designs should be developed to support videos that are compliant to the MPEG standards.

### B. Perceptual Encryption for JPEG Images

Due to the similarity between the encoding of JPEG images [26] and the frame-encoding of MPEG videos [24], [25], [27], the ideas of perceptual encryption for JPEG images can be easily extended to MPEG videos.

In [9], two techniques of perceptual encryption were studied: encrypting selective bit planes of uncompressed gray-scale images, and encrypting selective high-frequency ac coefficients of JPEG images, with a block cipher such as DES, triple-DES, or IDEA [28]. The continuous control of the visual quality degradation was not discussed, however.

In [10], the perceptual encryption of JPEG images is realized by encrypting variable-length codewords (VLCs) of partial ac coefficients in a zone of encryption (ZoE) to be other VLCs in the Huffman table. The visual quality degradation is controlled via an encryption probability,  $p/100 \in [0, 1]$ , where  $p \in \{0, \dots, 100\}$ . This encryption idea is similar to the VEA proposed in [22]. The main problem with encrypting VLCs is that the size of the encrypted image/video will be increased since the Huffman entropy compression is actually discarded in this algorithm.

### C. Perceptual Encryption for Wavelet-Compressed Images and Videos

In [11]–[13], several perceptual encryption schemes for wavelet-compressed images and videos were proposed. Under the control of a percentage ratio  $q$ , sign bit scrambling and secret permutations of wavelet coefficients/blocks/bit planes are combined to realize perceptual encryption. The problem with these perceptual encryption schemes is that the secret permutations are not sufficiently secure against known/chosen-plaintext attacks [29]–[32]: by comparing the absolute values of a number of plaintexts and ciphertexts, one can reconstruct the secret permutations. Once the secret permutations are removed, the encryption performance will be significantly compromised.

### D. Perceptual Encryption of Motion Vectors in MPEG-Videos

In [15], motion vectors are scrambled to realize perceptual encryption of MPEG-2 videos. Since I-frames do not depend on motion vectors, such a perceptual encryption algorithm can only blur the motions of MPEG videos. It cannot provide enough degradation of the visual quality of the MPEG videos for encryption (see Fig. 3). Generally speaking, this algorithm can be used as an option for further enhancing the performance of a perceptual encryption scheme based on other techniques.

### E. Pazarci–Dipçin Scheme

In [16], Pazarci and Dipçin proposed an MPEG-2 perceptual encryption scheme, which encrypts the video in the RGB color space via four secret linear transforms before the video is compressed by the MPEG-2 encoder. To encrypt the RGB-format uncompressed video, each frame is divided into  $M \times M$  scrambling blocks (SB), which is composed of multiple macroblocks of size  $16 \times 16$ . Assuming the input and the output pixel values are  $x_i$  and  $x_o$ , respectively, the four linear transforms are described as follows:

$$x_o = \begin{cases} \alpha x_i, & (D, N) = (0, 0) \\ FS - \alpha x_i, & (D, N) = (0, 1) \\ FS(1 - \alpha) + \alpha x_i, & (D, N) = (1, 0) \\ FS - [FS(1 - \alpha) + \alpha x_i], & (D, N) = (1, 1) \end{cases} \quad (1)$$

where  $\alpha = \alpha^*/100$  ( $\alpha^* \in \{50, \dots, 90\}$ ) is a factor controlling the visual quality degradation,  $D, N$  are two binary parameters that determine an affine transform for encryption, and  $FS$  means the maximal pixel value (for example,  $FS = 255$  for 8-bit RGB-videos). The value of  $\alpha$  in each SB is calculated from the preceding I-frame, with a function called  $\alpha$ -rule (see [16, Sec. 2.2] for more details). The  $\alpha$ -rule and its parameters are designated to be the secret key of this scheme.

The main merit of the Pazarci–Dipçin scheme is that the encryption/decryption and the MPEG encoding/decoding processes are separated, which means that the encryption part can simply be added to an MPEG system without any modification. However, the following defects make this scheme problematic in real applications.

- 1) Unrecoverable quality loss caused by the encryption always exists, unless  $\alpha = 1$  (which corresponds to no encryption). Even authorized users who know the secret key cannot recover the video with the original quality. Although it is claimed in [16] that human eyes are not

sensitive to such a quality loss if  $\alpha$  is set above 0.5, it may still be undesirable for high-quality video services, such as DVD and HDTV. In addition, limiting the value of  $\alpha$  lowers the security and flexibility of the encryption scheme.

- 2) The compression ratio may be significantly influenced by encryption if there are fast motions in the plain videos. This is because the motion compensation algorithm may fail to work for encrypted videos. The main reason is that the corresponding SBs may be encrypted with different parameters. To reduce this kind of influence, the encryption parameters of all SBs have to be sufficiently close to each other. This, however, compromises the encryption performance and the security.
- 3) The scheme is not suitable for encrypting MPEG-compressed videos. In many applications, such as VoD services, the plain-videos have already been compressed in MPEG format and stored in digital storage media (DSM). In this case, the Pazarci–Dipçin scheme becomes too expensive and slow, since the videos have to be first decoded, then encrypted, and finally encoded again. Note that the reencoding may reduce the video quality, since the encoder is generally different from the original one that produced the videos in the factory. Apparently, this defect is a natural side effect of the merit of the Pazarci–Dipçin scheme.
- 4) The scheme is not secure enough against brute-force attacks. For a given color component  $C$  of any  $2 \times 2$  SB structure, one can exhaustively guess the  $\alpha$ -values of the four SBs to recover the  $2 \times 2$  SB structure, by minimizing the block artifacts occurring between adjacent SBs. For each color component of a SB, the value of  $\alpha^* = 100\alpha \in \{50, \dots, 90\}$ ,  $D \in \{0, 1\}$  and  $N$  is determined by  $D$ , so one can calculate that the searching complexity is only  $(41 \times 2)^4 \approx 2^{25.4}$ , which is sufficiently small for PCs.<sup>1</sup> Once the value of  $\alpha$  of an SB is obtained, one can further break the secret key of the corresponding  $\alpha$ -rule. For the exemplified  $\alpha$ -rule given in [16, Eq. (3)], the secret key consists of the addresses of two selected subblocks (of size  $P \times P$ ) in a  $2 \times 2$  SB structure, and a binary shift value  $\Delta \in \{5, 50\}$ . Because  $\Delta$  can be uniquely determined from  $D$ , one only needs to search other part of the key, which corresponds to a complexity of  $\left(3 \times (2M/P)^2\right)^2$ . When  $P = M/2$ , the complexity is  $48^2 = 2304 \approx 2^{11.2}$ , and when when  $P = M/4$  it is  $192^2 = 36864 \approx 2^{15.2}$ . Apparently, the key space is not sufficiently large to resist brute-force attacks, either. In addition, since the values of quality factors and the secret parameters corresponding to the three color components can be separately guessed, the whole attack complexity is only three times of the above values, which is still too small from a cryptographical point of view [28]. Although using multiple secret keys for different SBs can increase the attack complexity exponentially, the key size will be too long and the key-management will become more complicated. Here, note that the  $\alpha$ -rule itself should not be considered as part of the key, following

the well-known Kerckhoffs' principle in modern cryptography [28].

- 5) The scheme is not sufficiently sensitive to the mismatch of the secret key, since the encryption transforms and the  $\alpha$ -rule given in [16] are both linear functions. This means that the security against brute-force attacks will be further compromised, as an approximate value of  $\alpha$  may be enough to recover most visual information in the plain-video.
- 6) The scheme is not secure enough against known/chosen-plaintext attacks. This is because the value of  $\alpha$  can be derived approximately from the linear relation between the plain pixel-values and the cipher pixel-values in the same SB. Similarly, the value  $N$  can be derived from the sign of the slope of the linear map between  $x_i$  and  $x_o$ , and the value of  $D$  can be derived from the value range of the map. Furthermore, assuming that there are  $k$  secret parameters in the  $\alpha$ -rule, if more than  $k$  different values of  $\alpha$  are determined as above, it is possible to uniquely solve the approximate values of the  $k$  secret parameters. To resist known/chosen-plaintext attacks, the secret key has to be changed more frequently than that suggested in [16] (one key per program), which will increase the computational burden of the servers (especially the key-management system).

#### F. Wang–Yu–Zheng Scheme

A different scheme working in the discrete cosine transform (DCT) domain (between DCT transform and Huffman entropy coding) was proposed by Wang, Yu, and Zheng in [17], which can be used as an alternative solution to overcome the first two shortcomings of the Pazarci–Dipçin scheme. By dividing all 64 DCT coefficients of each  $8 \times 8$  block into 16 subbands following the distance between each DCT coefficient and the dc coefficient, this new scheme encrypts the  $j$ th ac coefficient in the  $i$ th subband as follows:

$$b'_{ij} = \begin{cases} b_{ij} - \lfloor \beta a_i \rfloor, & b_{ij} \geq 0 \\ b_{ij} + \lfloor \beta a_i \rfloor, & b_{ij} < 0 \end{cases} \quad (2)$$

where  $b_{ij}$  and  $b'_{ij}$  denotes the plain pixel-value and the cipher pixel-value, respectively,  $\beta \in [0, 1]$  is the control factor,  $a_i$  is the rounding average value of all ac coefficients in the  $i$ th subband, and  $\lfloor \cdot \rfloor$  means the rounding function towards zero. The dc coefficients are encrypted in a different way, as  $b'_0 = b_0 \pm \lfloor C\beta a_0 \rfloor$ , where  $a_0 = b_0$  and  $C \in [0, 1]$  is the second control factor.<sup>2</sup> The value of  $a_i$  can also be calculated in a more complicated way to enhance the encryption performance, following [17, Eqs. (5) and (6)], where three new parameters,  $k_1, k_2, k_3$  are introduced to determine the values of  $a_i$  for the three color components, Y, Cr, and Cb. The 16 average values,  $a_0 \sim a_{15}$ , the two control factors,  $\beta$  and  $C$ , and the three extra parameters (if used),  $k_1, k_2, k_3$ , altogether serve as the secret scrambling parameters (i.e., the secret key) of each SB. Three different ways are suggested for the transmission of the secret parameters: 1) encrypting them and transmitting them in the payload of transport

<sup>1</sup>Even when  $\alpha^* \in \{0, \dots, 100\}$ , the searching complexity is only  $(101 \times 2)^4 \approx 2^{30.6}$ , which is still practically small.

<sup>2</sup>Note that the rounding function is missed in [17, Eqs. (3) and (4)]. In addition, [17, Eq. (4)] should read  $b'_0 = b_0 \pm \lfloor C\beta a_0 \rfloor$ , not  $b'_0 = a_0 \pm C\beta$ .

stream (TS); 2) embedding them in the high-frequency DCT coefficients; and 3) calculating them from the previous I-frame in a way similar to the  $\alpha$ -rule in [16].

In fact, the Wang–Yu–Zheng scheme is just an enhanced version of the Pazarci–Dipçin scheme, without amending all shortcomings of the latter scheme. Precisely, the following problems still remain.

- 1) Though the reduction of the compression ratio about motion compensations is avoided, the encryption will change the natural distribution of the DCT coefficients and thus reduce the compression efficiency of the Huffman entropy encoder. For example, when each subband has only one nonzero coefficient, it is possible that all 64 coefficients become nonzero after the encryption. This significantly increases the video size. In addition, if the secret parameters are embedded into the high-frequency DCT coefficients for transmission, the compression performance will be further compromised.
- 2) The scheme is still not sufficiently sensitive to the mismatch of the secret parameters, since the encryption function and the calculation function of  $a_i$  are kept linear. It is still not sufficiently secure against brute-force attacks to the secret parameters, because of the limited values of  $a_i, \beta, C, k_1, k_2, k_3$ . Furthermore, due to the nonuniform distribution of the DCT coefficients in each subband, an attacker needs not to randomly search all possible values of  $a_i$ .
- 3) This scheme is still insecure against known/chosen-plaintext attacks if the third way is used for calculating the secret parameters. In this case,  $a_i$  of each SB can be easily calculated from the previous I-frame of the plain-video. Additionally, since the value of  $\lfloor \beta a_i \rfloor$  can be obtained from  $b_{ij} - b'_{ij}$ , the secret parameter  $\beta$  can be derived approximately. In a similar way, the secret parameter  $C$  can also be derived approximately. If  $a_i$  is calculated with  $k_1, k_2, k_3$ , the values of  $\beta, k_1, k_2, k_3$  can be solved approximately with a number of known/chosen ac coefficients in four or more different subbands, so that  $C$  can be further derived from one known/chosen dc coefficient.
- 4) The method of transmitting the secret parameters in the payload of the transport stream cannot be used under the following conditions: 1) the key-management system is not available; and 2) the video is not transmitted with the TS format. A typical example is the perceptual encryption of MPEG-video files in personal computers.

### III. MORE EFFICIENT DESIGN OF PERCEPTUAL MPEG-VIDEO ENCRYPTION SCHEMES

Based on the analysis given above, we propose a simpler design of perceptual encryption for MPEG videos, and attempt to overcome the problems in existing schemes. The following useful features are supported in our new design.

- *Format-compliance*: the encrypted video can still be decoded by any standard-compliant MPEG decoder. This is a basic feature of all perceptual encryption schemes.
- *Lossless visual quality*: the encrypted video has the same visual quality as the original one, i.e., the original full-

quality video can be exactly recovered when the secret key is presented correctly.

- *Strict size-preservation*: the size of each data element in the lowest syntax level, such as VLCs, fixed-length codewords (FLCs), and continuous stuffing bits, remains unchanged after encryption. When the video stream is packetized in a system stream (i.e., PS or TS), the size of each video packet remains unchanged after encryption. This enables the following useful features in applications:
  - *independence of bit rate types (VBR and FBR)*;
  - *avoiding some time-consuming operations when encrypting MPEG-compressed videos*: bit rate control, repacketization of the system stream and the remultiplexing of multiple audio/video streams;
  - *on-the-fly encryption*: 1) direct encryption of MPEG-compressed video files without creating temporary files, i.e., one can open an MPEG video file, read the bit stream and simultaneously update (encrypt) it; 2) instantaneous switching encryption on/off for online video transmission;
  - *ROI (region of interest) encryption*: selectively encrypting partial frames, slices, macroblocks, blocks, motion vectors and/or DCT coefficients within specific regions of the video.
- *Independence of optional data elements in MPEG videos*: the perceptibility is not obtained by encrypting optional data elements, such as `quantiser_matrix` and `coded_block_pattern`.<sup>3</sup> This means that the scheme can encrypt any MPEG-compliant videos with a uniform performance.
- *Fast encryption speed*: 1) the extra computational load added by the encryption is much smaller than the computational load of a typical MPEG encoder; 2) MPEG-compressed videos can be quickly encrypted without being fully decoded and reencoded (at least the time-consuming inverse DCT (IDCT)/DCT operations are avoided).
- *Easy implementation*: the encryption/decryption parts can be easily incorporated into the whole MPEG system, without major modification of the structure of the codec.
- *Multidimensional perceptibility*: the degradation of visual quality is controlled by multidimensional factors.
- *Security against known/chosen-plaintext attacks* is ensured by four different measures.

To the best of our knowledge, some of the above features (such as on-the-fly encryption) have never been discussed in the literature on video encryption, in spite of their usefulness in real applications.

With the above features, the perceptual encryption scheme becomes more flexible to fulfill different requirements of various applications. To realize the strict size-preservation feature, the encryption algorithm has to be incorporated into the MPEG encoder, i.e., the (even partial) separation of the cipher and the encoder is impossible. This is a minor disadvantage in some applications. However, if the redesign is sufficiently simple, it is worth doing so to get a better tradeoff between the overall performance and the easy implementation. In the case that the re-

<sup>3</sup>Strictly speaking, motion vectors are also optional elements in MPEG videos, so the scheme should not encrypt only motion vectors as did in [15].

design of the MPEG codec is impossible, for example, if the codec is secured by the vendor, a simplified MPEG codec can be developed for the embedding of the perceptual video cipher. Since the most time-consuming operations in a normal MPEG codec including DCT/IDCT and picture reconstruction, are excluded from the simplified MPEG codec, fast encryption speed and low implementation complexity of the whole system can still be achieved.

In the following text of this section, we describe the design principle along with different methods of providing security against known/chosen-plaintext attacks, and discuss several implementation issues.

### A. New Design

This design is a generalized version of VEA [23] for perceptual encryption, by selectively encrypting FLC data elements in the video stream. Apparently, encrypting FLC data elements is the most natural and perhaps the simplest way to maintain all needed features, especially the need for the strict size-preservation feature. The proposed scheme is named PVEA. Note that PVEA can also be considered as an enhanced combination of the encryption techniques for JPEG images proposed in [9] and [10] and the perceptual encryption of motion vectors [15].

There are three main reasons for selecting only FLC data elements for encryption.

- 1) As analyzed below, all existing VLC encryption algorithms cannot be directly used to provide a controllable degradation of the quality. New ideas have to be developed to adopt VLC encryption in perceptual encryption schemes.
  - *VLC encryption with different Huffman tables* [23], [33]–[38]: Since each VLC-codeword is a pair of (run, level), if a VLC-codeword is decoded to get an incorrect “run” value, then the position of all the following DCT coefficients will be wrong. As a result, the visual quality of the decoded block will be degraded in an uncontrollable way. Thus, it is difficult to find a factor to control such visual quality degradation. Moreover, if the Huffman tables do not keep the size of each VLC-entry as designated in [33]–[37], syntax errors may occur when an unauthorized user decodes an encrypted video. This means that the encryption cannot ensure the format compliance to any standard MPEG codecs.
  - *VLC-index encryption* [21], [22]: This encryption scheme can ensure format compliance, but still suffers from the uncontrollability of the visual quality degradation due to the same reason as above. Another weakness of VLC-index encryption is that it may influence the compression efficiency and bring overhead on video size.
  - *Shuffling VLC-codewords or RLE events before the entropy encoding stage* [22], [39]: This algorithm can ensure both the format compliance and the strict size-preservation. However, even exchanging only two VLC-codewords may cause a dramatic change of the DCT coefficients distribution of each block. So, this encryption algorithm cannot realize a slight degradation of the visual quality and fails to serve as an ideal candidate for perceptual encryption.

- 2) It is obvious that FLC encryption is the simplest way to achieve all the desired properties mentioned in the beginning of this section, especially to achieve format compliance, strict size-preservation and fast encryption **simultaneously**. For example, naive encryption<sup>4</sup> can realize strict size-preservation and fast encryption, but cannot ensure format compliance.
- 3) As will be seen below, using FLC encryption is sufficient to fulfill the needs of most real applications for perceptual encryption.

According to MPEG standards [24], [25], [27], the following FLC data elements exist in an MPEG-video bit stream:

- 4-byte start codes: 000001xx (hexadecimal);
- almost all information elements in various headers;
- sign bits of nonzero DCT coefficients;
- (differential) dc coefficients in intra blocks;
- ESCAPE DCT coefficients;
- sign bits and residuals of motion vectors.

To maintain the format-compliance to the MPEG standards after the encryption, the first two kinds of data elements should not be encrypted. So, in PVEA, only the last four FLC data elements are considered, which are divided into three categories according to their contributions to the visual quality:

- *intra dc coefficients*: corresponding to the rough view (in the level of  $8 \times 8$  block) of the video;
- *sign bits of nonintra dc coefficients and ac coefficients, and ESCAPE DCT coefficients*: corresponding to details in  $8 \times 8$  blocks of the video;
- *sign bits and residuals of motion vectors*: corresponding to the visual quality of the video related to the motions (residuals further corresponds to the details of the motions).

Based on the above division, three control factors,  $p_{sr}$ ,  $p_{sd}$ , and  $p_{mv}$  in the range [0, 1], are used to control the visual quality in three different dimensions: the low-resolution rough (spatial) view, the high-resolution (spatial) details, and the (temporal) motions. With the three control factors, the encryption procedure of PVEA can be described as follows:

- 1) encrypting intra dc coefficients with probability  $p_{sr}$ ;
- 2) encrypting sign bits of nonzero DCT coefficients (except for intra dc coefficients) and ESCAPE DCT coefficients with probability  $p_{sd}$ ;
- 3) encrypting sign bits and residuals of motion vectors with probability  $p_{mv}$ .

The encryption of selected FLC data elements can be carried out with either a stream cipher or a block cipher. When a block cipher is adopted, the consecutive FLC data elements should be first concatenated together to form a longer bit stream, then each block of the bit stream is encrypted, and finally each encrypted FLC data element is placed back into its original position in the video stream. Under the assumption that the stream cipher or block cipher embedded in PVEA is secure, some special considerations should be taken into account in order to ensure the security against various attacks, as discussed below.

<sup>4</sup>In the image/video encryption literature, the term “naive encryption” means to consider the video as a 1-D bit stream and encrypt it via a common cipher.

In the above-described PVEA, the three factors control the visual quality, as follows:

- $p_{sr} = 1 \rightarrow 0$ : the spatial perceptibility changes from “almost imperceptible” to “perfectly perceptible” when  $p_{sd} = 0$  or to “roughly perceptible” when  $p_{sd} > 0$ ;
- $p_{sr} = 0, p_{sd} = 1 \rightarrow 0$ : the spatial perceptibility changes from “roughly perceptible” to “perfectly perceptible”;
- $p_{mv} = 1 \rightarrow 0$ : the temporal (motion) perceptibility (for P/B-pictures only) changes from “almost imperceptible” to “perfectly perceptible.”

The encryption may bring the recovered motion vectors out of the spatial range of the picture, so the motion compensation operations (or even the involved picture itself) may be simply discarded by the MPEG decoder. In this case, the temporal (motion) perceptibility will be “perfectly imperceptible,” not just “almost imperceptible.”

In [23, Appendix], it was claimed that the dc coefficients of each block can be uniquely derived from the other 63 ac coefficients. This means that the perceptual encryption of dc coefficients must not be used alone, i.e., some ac coefficients must also be encrypted to make the encryption of the dc coefficients secure. It was lately observed that this claim is not correct [40]. In fact, the dc coefficient of a block means the average brightness of the block, and is independent of the other 63 ac coefficients. Thus, the dc encryption and ac encryption of PVEA are independent of each other, i.e., the two control factors,  $p_{sr}$  and  $p_{sd}$ , are independent of each other, and they can be freely combined in practice.

### B. Security Against Ciphertext-Only Attacks and a Constraint of the Control Factor

The format compliance of perceptual encryption makes it possible for the attacker to guess the values of all encrypted FLC data elements separately in ciphertext-only attacks. The simplest attack is to try to recover more visual information by setting all the encrypted FLC data elements to zeros. This is called error-concealment-based attack (ECA) [22]. Our experimental results have shown that PVEA is secure against such attack. More details are given in the next section.

To guess the value of each FLC data element, one can also employ the local correlation existing between adjacent blocks in each frame. That is, one can search for a set of all encrypted FLC data elements in each frame to achieve the least blocking artifact. Does such a deblocking attack work? Now let us try to get a lower bound of this attack’s complexity, by assuming that the number of all FLC data elements in each frame is  $N$ , which means that the number of encrypted FLC data elements is  $pN$ . Then, the complexity of the deblocking attack will not be less than  $O\left(\binom{N}{pN}2^{pN}\right)$ , since each FLC data elements has at least two candidate values. So, if  $\binom{N}{pN}2^{pN}$  is cryptographically large, the deblocking attack will not compromise the security of PVEA. As a lower bound of  $p$  corresponding to a typical security level, one can get  $p \geq 100/N$  by assuming  $2^{pN} \geq 2^{100}$ . For most consumer videos that need to be protected via perceptual encryption,  $N$  is generally much larger than 100, so this constraint on  $p$  generally does not have too much influence on the overall performance of PVEA. Because the complexity  $2^{pN}$

is much over-estimated,<sup>5</sup> the constraint can be further relaxed in practice. For example, when  $N = 200$ , the above condition suggests that  $p \geq 100/N = 1/2$ . However, calculations showed that  $p \geq 9/100$  is enough to ensure a complexity larger than  $O(2^{100})$ .

Since it is generally impractical to carry out the deblocking attack on the whole frame, another two-layer deblocking attack may be adopted by the attacker: 1) performing the deblocking attack on small areas of the frame; and 2) for all candidates of these small areas, performing the deblocking attack on the area-level again. Though this two-layer attack generally has a much smaller complexity than the simple attack, its efficiency is still limited due to the following reasons.

- For each small area, the number of encrypted FLC elements is generally not equal to  $pN^*$ , where  $N^*$  denotes the total number of all FLC elements in the area. Thus, even this number has to be exhaustively guessed and then validated by considering the numbers of other areas (i.e., the whole frame). The existence of three independent quality factors makes the attack even more complicated.
- For small each areas, the probability that the least deblocking result does not correspond to the real scene may not very small. Accordingly, the attacker has to mount a more loose deblocking attack, thus leading to a higher attacking complexity.
- Even for the smallest area of size  $16 \times 16$ , there are generally more than one hundred FLC elements (i.e.,  $N^* \geq 100$ ), especially when there are rich visual information included in the area.
- If the number of FLC elements in an area is relatively small, this area generally contains less significant visual information (such as a smooth area).
- The smaller each area is, the more the number of fake results will be, and then the more the complexity of the second stage will be.

Of course, with the two-layer deblocking attack, the attacker can have a chance to recover a number of small areas, though he/she generally cannot get the whole frame. Such a minor security problem is an unavoidable result of the inherent format-compliance property of the perceptual encryption algorithms and related to the essential disadvantage of perceptual encryption exerted on some special MPEG-videos (see the discussion on Fig. 5 in the next section).

### C. Security Against Known/Chosen-Plaintext Attacks

Generally speaking, there are four different ways to provide security against known/chosen-plaintext attacks. Users can select one solution for a specific application.

1) *Using a Block Cipher*: With a block cipher, it is easy to provide security against known/chosen-plaintext attacks. Since the lengths of different FLC data elements are different, the block cipher may have to run in cipher feedback (CFB) mode with variable-length feedback bits to realize the encryption. Note that  $n$ -bit error propagation exists in block ciphers

<sup>5</sup>There are two reasons about the over-estimation: 1) the omission of  $\binom{N}{pN}$ , which is very large when  $N \gg pN$  and  $pN$  is not very small; 2) some FLC elements (such as intra dc coefficients) have more than 2 candidate values.

running in the CFB mode [28], where  $n$  is the block size of the cipher. It is also possible to cascade multiple FLC data elements to compose an  $n$ -bit block for encryption, as in RVEA [23, Sec. 7]. Compared to the CFB mode, the latter encryption mode can achieve a faster encryption speed (with a little more implementation complexity for bit cascading), since in the CFB mode only one element can be encrypted in each run of the block cipher.

2) *Using a Stream Cipher With Plaintext/Ciphertext Feedback:* After encrypting each plain data element, the plaintext or the ciphertext is sent to perturb the stream cipher for the encryption of the next plain data element. In such a way, the keystream generated by the stream cipher becomes dependent on the whole plain-video, which makes the known/chosen-plaintext attacks impractical. Note that an initial vector is needed for the encryption of the first plain data element.

3) *Using a Key-Management System and a Stream Cipher:* When a key-management system is available in an application, the encryption procedure of PVEA can be realized with a stream cipher. To effectively resist known/chosen-plaintext attacks, the secret key of the stream cipher should be frequently changed by the key-management system. In most cases, it is enough to change one key per picture, or per GOP. Note that this measure needs more computational load with higher implementation cost, and is suitable mainly for encrypting online videos.

4) *Using a Stream Cipher With UID:* When key-management systems are not available in some applications, a unique ID (UID) can be used to provide the security against known/chosen-plaintext attacks by ensuring that the UIDs are different for different videos. The UID of an MPEG-video can be stored in the user\_data area. The simplest form of the UID is the vendor ID plus the time stamp of the video. It is also possible to determine the UID of a video with a hash function or a secure pseudo-random number generator (PRNG). In this case, the UIDs of two different videos may be identical, but the probability is cryptographically small if the UID is sufficiently long. The UID is used to initialize the stream cipher together with the secret key, which ensures that different videos are encrypted with different keystreams. Thus, when an attacker successfully gets the keystream used for  $n$  known/chosen videos, he cannot use the broken keystreams to break other different videos. Of course, the employed stream cipher should be secure against plaintext attacks in the sense that the secret key cannot be derived from a known/chosen segment of the long keystream that encrypts the whole video stream [28].

#### D. Implementation Issues

Since PVEA is a generalization of VEA, it is obvious that fast encryption speed can be easily achieved, as shown in [23]. In addition, by carefully optimizing the implementation, the encryption speed can be further increased. We give two examples to show how to optimize the implementation of PVEA so as to increase the encryption speed.

A typical way to realize the probabilistic quality control with a decimal factor  $p$  is as follows: generate a pseudo-random decimal,  $r \in [0, 1]$ , for each data element with a uniformly-distributed PRNG, and then encrypt the current element only when

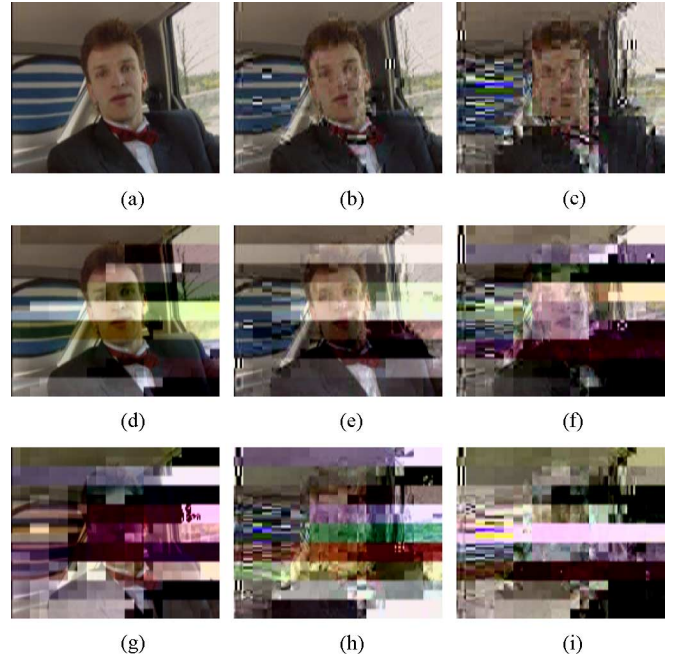


Fig. 2. Encryption results of the first frame in “Carphone.” (a)  $(p_{sr}, p_{sd}) = (0, 0)$ : the plain frame. (b)  $(p_{sr}, p_{sd}) = (0, 0.2)$ . (c)  $(p_{sr}, p_{sd}) = (0, 1)$ . (d)  $(p_{sr}, p_{sd}) = (0.2, 0)$ . (e)  $(p_{sr}, p_{sd}) = (0.2, 0.2)$ . (f)  $(p_{sr}, p_{sd}) = (0.5, 0.5)$ . (g)  $(p_{sr}, p_{sd}) = (1, 0)$ . (h)  $(p_{sr}, p_{sd}) = (1, 0.2)$ . (i)  $(p_{sr}, p_{sd}) = (1, 1)$ .



Fig. 3. Encryption results of the 313th frame in “Carphone.” (a)  $(p_{sr}, p_{sd}, p_{mv}) = (0, 0, 0)$ : the plain frame. ; (b)  $(p_{sr}, p_{sd}, p_{mv}) = (0, 0, 0.5)$ . (c)  $(p_{sr}, p_{sd}, p_{mv}) = (0, 0, 1)$ .

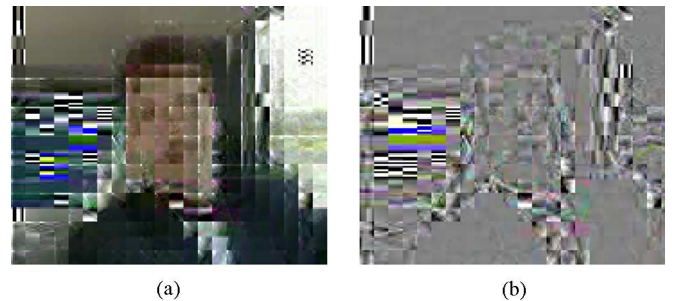


Fig. 4. Recovered results after applying ECA for the first frame in “Carphone.” (a) Breaking Fig. 2(c). (b) Breaking Fig. 2(i).

$r \leq p$ . The above implementation can be modified as follows to further increase the encryption speed:

- 1) pseudo-randomly select  $N_p = \text{round}(N \cdot p)$  integers from the set  $\{0, \dots, N - 1\}$ ;
- 2) create a binary array  $SE[0] \sim SE[N]$ :  $SE[i] = 1$  if the integer  $i$  is selected; otherwise,  $SE[i] = 0$ ;
- 3) encrypt the  $i$ th FLC data element only when  $SE[i \bmod N] = 1$ .

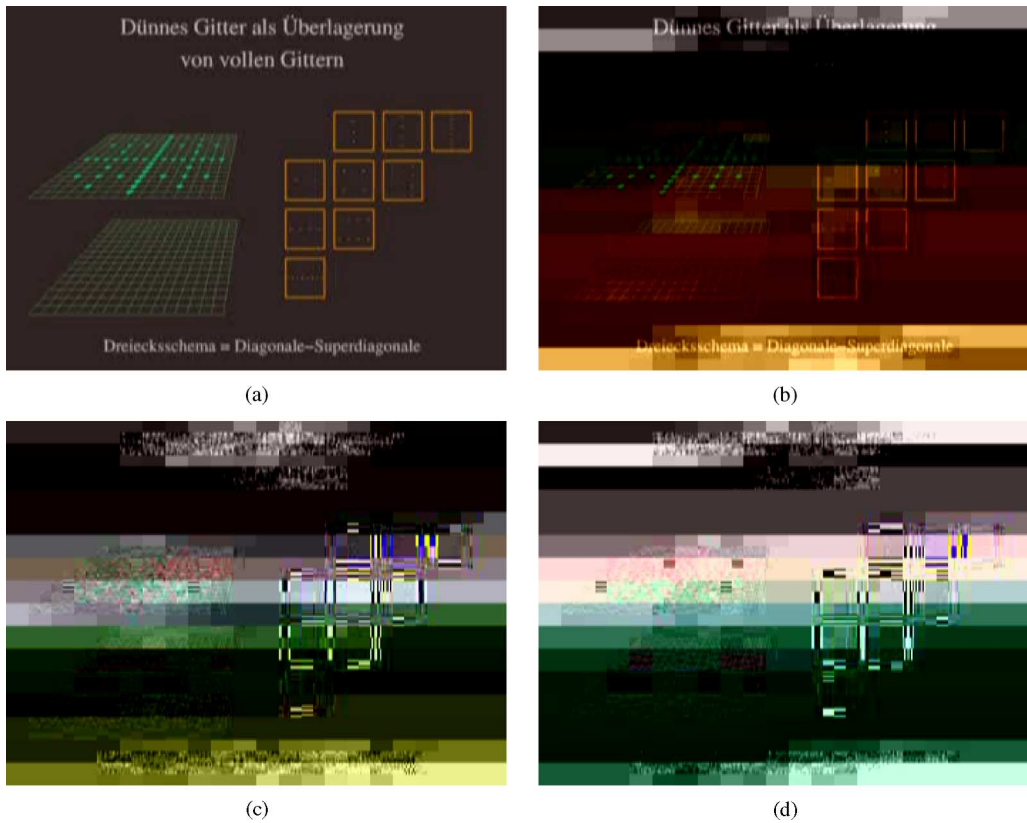


Fig. 5. Encryption results of the 169th frame in an MPEG-1 video. (a)  $(p_{sr}, p_{sd}, p_{mv}) = (0, 0, 0)$ : the plain frame. (b)  $(p_{sr}, p_{sd}, p_{mv}) = (1, 0, 0)$ . (c)  $(p_{sr}, p_{sd}, p_{mv}) = (1, 1, 0)$ . (d)  $(p_{sr}, p_{sd}, p_{mv}) = (1, 1, 1)$ .

In this modified implementation, only a modulus addition and a look-up-table operation are needed to determine whether the current data element should be encrypted. As a comparison, in the typical implementation, one run of the PRNG is needed for each data element, which is generally much slower. Although  $N$  bits of extra memory is needed to store the array in the modified implementation, it is merely a trivial problem since video codec generally requires much more memory. To ensure the security against deblocking attacks, in the modified implementation the value of  $N$  should not be too small.<sup>6</sup>

To further reduce the computational load of PVEA, another way is to selectively encrypt partial FLC data elements. Two possible options are as follows: 1) encrypt only intra blocks; and 2) encrypt only sign bits (or a few number of most significant bits) of intra dc coefficients, ESCAPE DCT coefficients, and residuals of motion vectors. The above two options can also be combined together. This will have very little effect on the encryption performance, since an attacker can only recover video frames with a poor visual quality from other unencrypted data elements [22], [41].

#### IV. ENCRYPTION PERFORMANCE OF PVEA

Some experiments have been conducted to test the real encryption performance of PVEA for a widely-used MPEG-1 test video, “Carphone.” The encryption results of the 1st frame (I-type) are shown in Fig. 2, with different values of the two

control factors  $p_{sr}$  and  $p_{sd}$ . It can be seen that the degradation of the visual quality is effectively controlled by the two factors. The encryption results of the third control factor  $p_{mv}$  are given in Fig. 3, where the 313th frame (B-type) is selected for demonstration. It can be seen that encrypting only the motion vectors will not cause much degradation in the visual quality.

Our experiments have also shown that PVEA is secure against error-concealment based attacks. For two encrypted frames shown in Fig. 2, the recovered images after applying ECA are shown in Fig. 4. In Fig. 4(a), the sign bits of all ac coefficients are set to be zeros, and in Fig. 4(b) all dc coefficients are also set to be zeros. It can be seen that the visual quality of the recovered images via such an attack is even worse than the quality of the cipher-images, which means that ECA cannot help an attacker get more visual information. Actually, the security of PVEA against ECA depends on the fact that an attacker cannot tell encrypted data elements from un-encrypted ones without breaking the key. As a result, he has to set all possible data elements to be fixed values, which is equivalent to perceptual encryption with the control factor 1, i.e., the strongest level of perceptual encryption.

Finally, it is worth mentioning that PVEA has a minor disadvantage that the degradation in the visual quality is dependent on the amplitudes of the intra dc coefficients. As an extreme example, consider an intra picture whose dc coefficients are all zeros, which means that the FLC-encoded differential value of each intra dc coefficient does not occur in the bit stream, i.e., only the VLC-encoded  $\text{dct\_dc\_size} = 0$  occurs. In this case,

<sup>6</sup>In most cases, it is enough to set  $N \geq 300$ .



the control of the rough visual quality by  $p_{sr}$  completely disappears. Similarly, when  $\text{dct\_dc\_size} = 1$ , the encryption can only change the differential value from  $\pm 1$  to  $\mp 1$ , so the degradation will not be very significant. As a result, this problem will cause the perceptibility of some encrypted videos become “partially perceptible” when  $p_{sr} = 1$  (should be “almost imperceptible” for most videos). For an MPEG-1 video<sup>7</sup> with a dark background (i.e., with many intra dc coefficients of small amplitudes), the encryption results are shown in Fig. 5. Fortunately, this problem is not so serious in practice, for the following reasons:

- most consumer videos contain sufficiently many intra dc coefficients of large amplitudes;
- even when there are many zero intra dc coefficients, the content of the video has to be represented by other intra dc coefficients of sufficiently large amplitudes;
- the differential encoding can increase the number of nonzero intra dc coefficients;
- the partial degradation caused by  $p_{sr}$  and the degradation caused by  $p_{sd}$  and  $p_{mv}$  are enough for most applications of perceptual encryption (see Figs. 2 and 5).

From this minor disadvantage of PVEA, a natural result can be immediately derived: for the protection of MPEG videos that are highly confidential, VLC data elements should also be encrypted. In fact, our additional experiments on various VEAs have shown that it might be impossible to effectively degrade the visual quality of the MPEG videos with dark background via format-compliant encryption, unless the compression ratio and the strict size-preservation feature are compromised. The relations among the encryption performance, the compression ratio, the size-preservation feature, and other features of VEAs, are actually much more complicated. These problems will be investigated in our future research.

## V. CONCLUSION

This paper focuses on the problem of how to realize perceptual encryption of MPEG videos. Based on a comprehensive survey on related work and performance analysis of some existing perceptual video encryption schemes, we have proposed a new design with more useful features, such as on-the-fly encryption and multidimensional perceptibility. We have also discussed its security against deblocking attack and pointed out some measures against known/chosen-plaintext attack. The proposed perceptual encryption scheme can also be extended to realize nonperceptual encryption by simply adding a VLC-encryption part.

## REFERENCES

- [1] W. Zeng, H. Yu, and C.-Y. Lin, *Multimedia Security Technologies for Digital Rights Management*. New York: Academic, 2006.
- [2] A. Uhl and A. Pommer, *Image and Video Encryption: From Digital Rights Management to Secured Personal Communication*. Boston, MA: Springer Science + Business Media, Inc., 2005, vol. 15, Advances in Information Security.
- [3] B. Furht, E. Muharemagic, and D. E. Socek, Eds., *Multimedia Encryption and Watermarking*. New York: Springer, 2005.
- [4] B. Furht, D. Socek, and A. M. Eskicioglu, “Fundamentals of multimedia encryption techniques,” in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds. Boca Raton, FL: CRC Press, 2004, ch. 3, pp. 93–131.
- [5] S. Li, G. Chen, and X. Zheng, “Chaos-based encryption for digital images and videos,” in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds. Boca Raton, FL: CRC Press, 2004, ch. 4, pp. 133–167.
- [6] A. Servetti and J. C. D. Martin, “Perception-based selective encryption of G.729 speech,” in *Proc. IEEE ICASSP'2002*, 2002, vol. 1, pp. 621–624.
- [7] —, “Perception-based partial encryption of compressed speech,” *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 637–643, Aug. 2002.
- [8] A. Torrubia and F. Mora, “Perceptual cryptography on MPEG layer III bit streams,” *IEEE Trans. Consum. Electron.*, vol. 48, no. 4, pp. 1046–1050, Apr. 2002.
- [9] M. V. Droogenbroeck and R. Benedett, “Techniques for a selective encryption of uncompressed and compressed images,” in *Proc. ACIVS'2002*, pp. 90–97.
- [10] A. Torrubia and F. Mora, “Perceptual cryptography of JPEG compressed images on the JFIF bit stream domain,” in *Proc. Dig. ICCE'2003*, pp. 58–59.
- [11] S. Lian, J. Sun, and Z. Wang, “Perceptual cryptography on SPIHT compressed images or videos,” in *Proc. IEEE ICME'2004*, pp. 2195–2198.
- [12] —, “Perceptual cryptography on JPEG2000 compressed images or videos,” in *Proc. Int. Conf. CIT'2004*, pp. 78–83.
- [13] S. Lian, X. Wang, J. Sun, and Z. Wang, “Perceptual cryptography on wavelet-transform encoded videos,” in *Proc. IEEE ISIMP'2004*, pp. 57–60.
- [14] J. Dittmann and A. Steinmetz, “Enabling technology for the trading of MPEG-encoded video,” in *Proc. ACISP'97*, vol. 1270, pp. 314–324.
- [15] Y. Bodo, N. Laurent, and J.-L. Dugelay, “A scrambling method based on disturbance of motion vector,” in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 89–90.
- [16] M. Pazarci and V. Dipçin, “A MPEG2-transparent scrambling technique,” *IEEE Trans. Consum. Electron.*, vol. 48, no. 2, pp. 345–355, Feb. 2002.
- [17] C. Wang, H.-B. Yu, and M. Zheng, “A DCT-based MPEG-2 transparent scrambling algorithm,” *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1208–1213, Apr. 2003.
- [18] A. Pommer and A. Uhl, “Selective encryption of wavelet packet sub-band structures for obscured transmission of visual data,” in *Proc. 3rd IEEE Benelux SPS'2002*, pp. 25–28.
- [19] —, “Selective encryption of wavelet-packet encoded image data: Efficiency and security,” *Multimedia Syst.*, vol. 9, no. 3, pp. 279–287, 2003.
- [20] A. Pommer, “Selective encryption of wavelet-compressed visual data,” Ph.D. dissertation, Dept. Sci. Comput., Univ. Salzburg, Salzburg, Austria, 2003.
- [21] J. Wen, M. Severa, W. Zeng, M. H. Luttrell, and W. Jin, “A format-compliant configurable encryption framework for access control of multimedia,” in *Proc. IEEE 4th Workshop MMSP'2001*, pp. 435–440.
- [22] —, “A format-compliant configurable encryption framework for access control of video,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 545–557, Jun. 2002.
- [23] B. Bhargava, C. Shi, and S.-Y. Wang, “MPEG video encryption algorithms,” *Multimedia Tools Appl.*, vol. 24, no. 1, pp. 57–79, 2004.
- [24] *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2 and ITU-T Rec. H.262 (MPEG-2), 2000.
- [25] *Information Technology – Coding of Audio-Visual Objects – Part 2: Visual*, ISO/IEC 14496-2 (MPEG-4), 2004.
- [26] *Information Technology – Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines*, ISO/IEC 10918-1 (JPEG), 1994.
- [27] *Information Technology – Coding of Moving Pictures and Associated Audio For Digital Storage Media at up to About 1.5 Mbit/s – Part 2: Video*, ISO/IEC 11172-2 (MPEG-1), 1993.
- [28] B. Schneier, *Applied Cryptography – Protocols, Algorithms, and Source Code in C*, 2nd ed. New York: Wiley, 1996.
- [29] J.-K. Jan and Y.-M. Tseng, “On the security of image encryption method,” *Inf. Process. Lett.*, vol. 60, no. 5, pp. 261–265, 1996.
- [30] C.-C. Chang and T.-X. Yu, “Cryptanalysis of an encryption scheme for binary images,” *Pattern Recognit. Lett.*, vol. 23, no. 14, pp. 1847–1852, 2002.

<sup>7</sup>Source of this test video: [http://www5.in.tum.de/forschung/visualisierung/duenne\\_gitter/DG\\_4.mpg](http://www5.in.tum.de/forschung/visualisierung/duenne_gitter/DG_4.mpg).

- [31] X.-Y. Zhao, G. Chen, D. Zhang, X.-H. Wang, and G.-C. Dong, "Decryption of pureposition permutation algorithms," *J. Zhejiang Univ. SCIENCE*, vol. 5, no. 7, pp. 803–809, 2004.
- [32] S. Li, C. Li, G. Chen, D. Zhang, and N. G. Bourbakis, A General Cryptanalysis of Permutation-Only Multimedia Encryption Algorithms Cryptology ePrint Archive, 2004, Tech. Rep. 2004/374.
- [33] C.-P. Wu and C.-C. J. Kuo, "Fast encryption methods for audiovisual data confidentiality," *Proc. SPIE Multimedia Syst. Appl. III*, vol. 4209, pp. 284–295, 2001.
- [34] C.-P. Wu and C.-C. J. Kuo, "Efficient multimedia encryption via entropy codec design," *Proc. SPIE Security Watermarking Multimedia Contents III*, vol. 4314, pp. 128–138, 2001.
- [35] D. Xie and C.-C. J. Kuo, "An enhanced MHT encryption scheme for chosen plaintext attack," *Proc. SPIE Internet Multimedia Manag. Syst. IV*, vol. 5242, pp. 175–183, 2003.
- [36] C.-P. Wu and C.-C. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Trans. Multimedia*, vol. 7, pp. 828–839, Jul. 2005.
- [37] M. S. Kankanhalli and T. T. Guan, "Compressed-domain scrambler/descrambler for digital video," *IEEE Trans. Consum. Electron.*, vol. 48, no. 2, pp. 356–365, Feb. 2002.
- [38] C. Shi and B. Bhargava, "Light-weight MPEG video encryption algorithm," in *Proc. Int. Conf. Multimedia*, 1998, pp. 55–61.
- [39] G. Liu, T. Ikenaga, S. Goto, and T. Baba, "A selective video encryption scheme for MPEG compression standard," *IEICE Trans. Fundamentals*, vol. E89-A, no. 1, pp. 194–202, 2006.
- [40] S. Li and B. Bhargava, A Note on MPEG Video Encryption Algorithms' 2004 [Online]. Available: [http://www.hooklee.com/Papers/MMTA2004\\_note.pdf](http://www.hooklee.com/Papers/MMTA2004_note.pdf)
- [41] I. Agi and L. Gong, "An empirical study of secure MPEG video transmission," in *Proc. ISOC SNDSS'96*, pp. 137–144.



computer identification.

**Shujun Li** received the B.S. degree in information science and engineering and the Ph.D. degree in information and communication engineering, both from the Xian Jiaotong University, Xian, China, in 1997 and 2003, respectively.

Currently he is a Postdoctoral Fellow with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. His current research interests include multimedia security (mainly image and video encryption), chaotic cryptography, and secure human-



**Guanrong Chen** (M'89–M'92–F'97) received the M.Sc. degree in computer science from Zhongshan University, Zhongshan, China, and the Ph.D. degree in applied mathematics from Texas A&M University, College Station.

Currently he is a Chair Professor and the Founding Director of the Centre for Chaos and Complex Networks, City University of Hong Kong. He has (co)authored 15 research monographs and advanced textbooks, more than 400 journal papers, and about 200 refereed conference papers, published since 1981 in

the fields of nonlinear system dynamics and controls. He is Honorary Professor of the Central Queensland University, Australia, and of more than ten universities in China.

Dr. Chen is a Fellow of the IEEE for his fundamental contributions to the theory and applications of chaos control and bifurcation analysis. He has served or is serving as Editor for eight international journals, including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and *International Journal of Bifurcation and Chaos*, and received four best journal paper awards in the past.



**Albert Cheung** (M'95) teaches electrical service engineering at the Building and Construction Department, City University of Hong Kong, Hnig Kong.

He works on sustainable energy, voice and video encryption, and electromechanical applications.



**Bharat Bhargava** (F'93) received the B.E. degree from the Indiana Institute of Science and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN.

He is currently a Professor of computer science at Purdue University. His research involves mobile wireless networks, host authentication and key management, secure routing and dealing with malicious hosts, adaptability to attacks, and experimental studies.

Dr. Bhargava has been awarded the Charter Gold Core Member distinction by the IEEE Computer Society for his distinguished service and received an IEEE technical achievement award for the major impact of his decade long contributions to the foundations of adaptability in communication and distributed systems in 1999. He is a fellow of the IEEE Computer Society. His name has been included in the Book of Great Teachers at Purdue and was selected by the student chapter of ACM at Purdue for the Best Teacher Award.



**Kwok-Tung Lo** (M'92) was born in Hong Kong. He received the M.Phil. and Ph.D. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, in 1989 and 1992, respectively.

Since 1992, he has been with the Hong Kong Polytechnic University, Hong Kong, where he is now an Associate Professor at the Department of Electronic and Information Engineering. He is very active in research and has published over 130 papers in various international journals and conference proceedings. He is one of the authors of the book

*Fundamentals of Image Coding and Wavelet Compression: Principles, Algorithms and Standards* (Tsinghua Univ. Press). He is currently a member of the Editorial Board of Multimedia Tools and Applications and an Associate Editor of HKIE Transactions. His current research interests include multimedia signal processing, digital watermarking, multimedia communications and Internet applications.