Multimedia Systems
DOI 10.1007/s00530-007-0083-z

REGULAR PAPER

New approaches to encryption and steganography for digital videos

Daniel Socek · Hari Kalva · Spyros S. Magliveras · Oge Marques · Dubravko Culibrk · Borko Furht

Abstract In this work we propose a novel type of digital video encryption that has several advantages over other currently available digital video encryption schemes. We also present an extended classification of digital video encryption algorithms in order to clarify these advantages. We analyze both security and performance aspects of the proposed method, and show that the method is efficient and secure from a cryptographic point of view. Even though the method is currently feasible only for a certain class of video sequences and video codecs, the method is promising and future investigations might reveal its broader applicability. Finally, we extend our approach into a novel type of digital video steganography where it is possible to disguise a given video with another video.

1 Introduction

The security and privacy of digital videos has become increasingly more important in today's highly computerized and interconnected world. Digital media content must be protected in applications such as pay-per-view TV or confidential video conferencing, as well as in medical, industrial or military multimedia systems. With the rise of wireless portable devices, many users seek to protect the private multimedia messages that are exchanged over the wireless or wired networks. In general, applying a well-established, general-purpose symmetric-key encryption algorithm to ensure the confidentiality during video transmission is a good idea from a security point of view. Unfortunately, conventional, general-purpose encryption algorithms, such as AES, are not suitable for a number of digital video applications [6,7,11,12,15,20,28], mainly since these algorithms do not conform to various video application-related requirements, which are discussed in Sect. 2. In order to overcome this problem, a significant number of video encryption algorithms specifically designed for digital videos have been proposed [1,16,17,20,24,31]. Even though several classifications of video encryption algorithms have been previously presented [12,15], we provide an extended and more comprehensive such classification. Furthermore, we present and analyze a video encryption mechanism that falls into a new category of video encryption algorithms. An algorithm from this category inherently possesses several advantages over other schemes. Finally, we show that our method allows for a new type of digital video steganography where a given video is disguised with another video.

The rest of this paper is organized as follows. In the next section we review some of the application-related requirements that are not addressed with the conventional cryptography. Section 3 provides an extended comprehensive classification of video encryption algorithms, with a brief survey of the relevant published research in the area of video encryption. Our novel approach is presented in Sect. 4, with its security analysis, implementation issues, and performance analysis given in Sects. 5, 6 and 7, respectively. A new approach to video steganography is presented in Sect. 8. Finally, Sect. 9 holds our conclusions and suggestions for further research.

D. Socek (✉) · H. Kalva · O. Marques · D. Culibrk · B. Furht
Department of Computer Science and Engineering,
Florida Atlantic University, Boca Raton, FL 33431, USA
e-mail: dsocek@fau.edu

S. S. Magliveras
Department of Mathematical Sciences,
Florida Atlantic University, Boca Raton, FL 33431, USA

2 Application-related requirements for video encryption algorithms

There are applications with requirements not supported by the conventional encryption methods. Thus, the encryption algorithms specifically designed to support these requirements are desirable. These requirements include the following:

1. Perceptual quality control. Encryption methods could be used to intentionally degrade the quality of perception. If the video contains sensitive industrial, governmental or military information, then the cryptographic strength must be substantial (high-security) and no perceptual information should be preserved after encryption.
2. Format-compliance. In many applications it is desired that the encryption algorithm preserves the video compression format. In other words, after encrypting the encoded video, ordinary decoders can still decode it without crashing. This property of an encryption algorithm is often called *format-compliance* (also called *transparency*, *transcodability* or *syntax-awareness*). As Fig. 1 shows, ordinary decoders are able to process format-compliant encrypted data. Depending on a type of encryption used, the produced output appears either perceivable but distorted, or non-perceivable and random.
3. Codec standard-compliance. A method is codec-standard compliant if it conforms the used video codec standard. A typical video system is likely to consist of a pre-manufactured encoder and decoder modules, and a video encryption method that requires no modification to either of the two modules is often desirable.
4. Target bitrate. The amount of data that the video system should process per unit of time is referred to as the *target bitrate*. Typically, this is the function of the time used for encoding, encryption, transmission (which depends on the channel bandwidth), decoding, and decryption. In many real-time video applications, it is imperative that the speed of the encryption and decryption algorithms be fast enough to ensure the target bitrate that is usually needed for the normal video system processing.
5. Constant bitrate. In many instances, it is also required that the encryption transformation preserves the size of a bitstream. This is known as the *constant bitrate* requirement. However, more often than not, it is simply preferred that the output produced by an encryption-equipped encoder and the output produced by an ordinary encoder have similar sizes.
6. Error-tolerance. For many multimedia systems *error-tolerance*, or *error-resilience*, is usually of high importance. Most conventional ciphers possess strong avalanche property, which causes decryption to fail even if a single bit is flipped during transfer. Some video codecs (e.g. H.264) have their own error correcting mechanisms. Video encryption algorithm that preserves these mechanisms is favorable for video systems with noisy channels.

In general, modern cryptography is designed for a generic bitstream, and as such, it disregards the aforementioned properties of a digital video and the requirements of a typical digital video application. In the next section, we present a classification and a brief overview of the existing video encryption algorithms proposed in the past mainly to overcome some of these application-related issues.

3 Classification and overview of video encryption algorithms

Research in the area of digital video encryption represents a relatively new area of study, yet a considerable number of video encryption approaches have been proposed. In several publications, such as [15] and [12], the authors present a very limited classification regarding digital video encryption approaches taken by the research community. However, to date no comprehensive classification is provided in the published scientific literature. In this section we present an extended, comprehensive formal classification that includes some new concepts related to our approach. Having a comprehensive classification of digital video encryption algorithms in place helps the generalization and understanding of core aspects such as security issues, functionality objectives,

(a) **(b)** **(c)**

Fig. 1 Decoded video produced by an ordinary decoder for **a** video without encryption, **b** video encrypted with format-compliant perceptual encryption, and **c** video encrypted with a high-security format-compliant encryption

composability and applicability. Moreover, many features, properties and issues regarding the proposed approaches can be studied at the class level, which aids in the evaluation of existing approaches, as well as the design and evaluation of future proposals.

3.1 Classification criteria and taxonomy

In general, all digital video encryption algorithms can be classified according to the following seven classification criteria (C1–C7):

C1. What is selected for encryption?
C2. Is the content perceivable after encryption?
C3. Is the bitstream format-compliant after the encryption step?
C4. Does the approach require codec modification?
C5. How does the bitstream size fluctuate due to the encryption algorithm?
C6. Does the encryption occur before, during, or after the compression?
C7. Does the encryption process use a conventional cryptosystem?

Classification in terms of what is selected for encryption (C1):

Full-encryption approaches. A video e ncryption algorithm that performs encryption on the entire video bitstream (either compressed or uncompressed) belongs to this class of algorithms. This class of algorithms includes the so-called *naïve approach*. The naïve approach is a type of full encryption approach in which a conventional cryptosystem is used in the encryption step. All other full-encryption approaches that are not using a conventional, general-purpose cryptosystem are grouped into *non-conventional full-encryption approaches*. These non-conventional approaches are designed mostly to accommodate low computational complexity and fast performance. Chaos-based video encryption algorithms, as well as the approaches proposed in this dissertation are examples of such algorithms.

Selective (partial) encryption approaches. This class of video encryption algorithms consists of all approaches that perform encryption only on certain, carefully selected bits from the video bitstream (compressed or uncompressed), while leaving the rest of the bits unencrypted. Even more generally, one can define *variable encryption* to be an approach where different encryption security levels are applied to different bits from the input. Selective encryption (also called *partial encryption*) can be seen as a subcategory of variable encryption. A special type of selective encryption approach where the bits are selected based on spatial information is referred to as *spatially selective encryption*. For instance, one may choose to encrypt only the face of the person appearing in the video sequence.
Classification according to the degree of perception (C2):

Perceptual encryption approaches. As a quality control mechanism, it is often desirable to have a video encryption approach that intentionally preserves some low-quality perceptual information about the source video content. By decrypting such encrypted video, one gains access to the source video of original visual quality. In addition, being able to control the degree of perception with a choice of encryption "strength" is often desirable. Perceptual encryption approaches are inherently of low-security in terms of content confidentiality. However, they should be of high-security regarding the quality reconstruction control.

Non-perceivable encryption approaches. Approaches that do not preserve any perceptual information after encryption are associated with this class of video encryption algorithms. An algorithm targeted for high-security video applications should belong to this class of algorithms.
Classification in terms of the encrypted bitstream format-compliance (C3):

Format-compliant approaches. When an encrypted video content is broadcast, it is often desired that the encrypted bitstream is compliant with the appropriate video format so that the decoders at the client side would not crash. Video encryption approaches that preserve the video compression format after the encryption step belong to this class of algorithms.

Format-defiant approaches. As opposed to the format-compliant algorithms, *format-defiant algorithms* produce output that is not compatible with the appropriate video encoding format.
Classification in terms of codec standard-compliance (C4):

Codec standard-compliant approaches. Algorithms from this class do not require modification to either the encoder or the decoder. These algorithms are favorable for those systems where replacing the pre-installed software or hardware codecs would be infeasible. Full encryption approaches are associated with this group of algorithms.

Codec standard-defiant approaches. This class of video encryption algorithms consists of methods that require codec modification. Most of the selective encryption approaches belong to this category of algorithms, since generally they require modification to both the encoder and the decoder.
Classification according to the bitstream size fluctuation (C5):

Constant (or near-constant) bitrate approaches. Algorithms from this class do not interfere with the compression performance of the encoder. In general, the non-encrypted video and its encrypted version should always be of the same size (*constant bitrate*), or very close to one another (*near-constant bitrate*).

Variable bitrate approaches. This class of video encryption approaches consists of methods that produce a variable bitstream size. Video encryption algorithms that significantly increase the bitstream size are of little interest to any practical application. However, algorithms that have the ability to decrease, preserve, or at least tolerably increase the size of the bitstream are much more practical. The approaches proposed in this dissertation are from the latter class of algorithms.
Classification in terms of when the encryption occurs (C6):

In-compression approaches. Algorithms that perform encryption within the video encoder belong to this class. This implies that the decryption must occur within the decoding process at the decoder side. An algorithm from this class is inherently codec standard-defiant. Many selective video encryption algorithms belong to this group.

Post-compression approaches. Algorithms from this class perform the encryption step after the video has been compressed. A naïve approach, as well as many non-conventional full approaches, such as [16], are indeed from this group of algorithms. These approaches are inherently codec standard-compliant, but also inherently format-defiant.

Pre-compression approaches. This class of video encryption algorithms consists of approaches that perform encryption before the compression. The algorithms from this class are inherently format-compliant and codec standard-compliant.
Classification in terms of the underlying cryptosystem (C7):

Conventional encryption-based approaches. This class consists of approaches where a conventional encryption algorithm is used in the encryption step. Many selective approaches belong to this class. A full approach that belongs to this class is inherently a naïve approach.

Non-conventional encryption-based approaches. If the encryption process is based on a novel multimedia-specific type of encryption, the algorithm should be classified as non-conventional encryption-based. For example, approaches like Huffman table permutations, chaotic map-based or Hopfield neural network-based multimedia encryptions all belong to this class.

Several aforementioned observations regarding the presented classification should be summarized:

- Perceptual encryption approaches must be format-compliant.
- In-compression approaches are inherently codec standard-defiant.
- Post-compression approaches are inherently format-defiant.
- Pre-compression approaches are inherently format-compliant and codec standard-compliant.
- Full encryption approach that is also conventional encryption-based is by definition a naïve approach.

3.2 Overview of related work

The idea of selective video encryption was introduced independently by Meyer and Gadegast [17] with an algorithm *SECMPEG*, and by Spanos and Maples [24] with an algorithm *Aegis*. Since then, a significant number of selective encryption proposals appeared in the scientific literature. A good reference for selective image and video encryption methods along with some security analysis is given in [7, 8, 12, 15, 28]. The proposed selective video encryption approaches can generally be partitioned into two groups: (1) approaches that select information from the compressed bitstream (e.g. encrypting only compressed bitstream headers), and (2) approaches where the selection is performed during the compression step (e.g. encrypting DCT coefficients). Figure 2 shows the basic architecture of these two selective encryption techniques.

Many selective encryption approaches were successfully cryptanalyzed shortly after the initial proposal as scientists found ways to exploit the information from the remaining, unencrypted bits. For example, Agi and Gong [2] showed weaknesses in SECMPEG [17] and Aegis [24] a year later. Also, a year after Tang proposed an image/video encryption method [27] based on permuting the zig-zag reordering after a DCT transformation, Qiao et al. [21] discovered serious weaknesses against attacks derived from statistical analysis of the unencrypted DCT coefficients. Seidel et al. [25] show some weaknesses in the video encryption algorithms by Shi et al. shortly after the original proposals [3, 22, 26]. Due to complexity of the compressed video bitstream, it is often difficult to analyze how much the unencrypted bits can truly tell about the encrypted ones, or how can one use that information to aid cryptanalysis. Hence, the selective encryption approaches are much easier to design and evaluate against the performance aspects, than to properly evaluate in terms of security.

Selective approaches where the selection is performed during the compression step (as depicted in Fig. 2a) are inherently codec-standard defiant. Most selective encryption approaches, especially the ones that select information directly

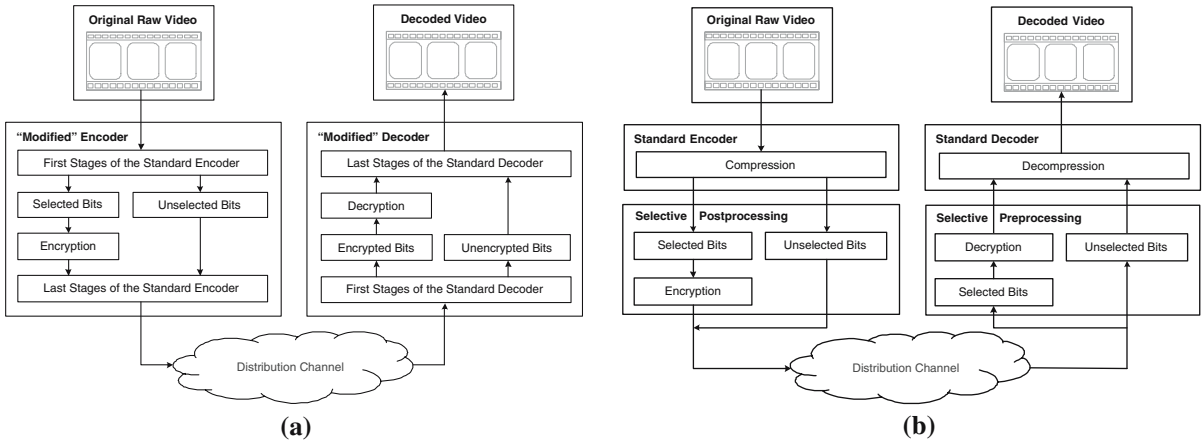

Fig. 2 The usual architecture for selective video encryption approaches where selection and encryption occurs: **a** during compression stage, and **b** after compression stage

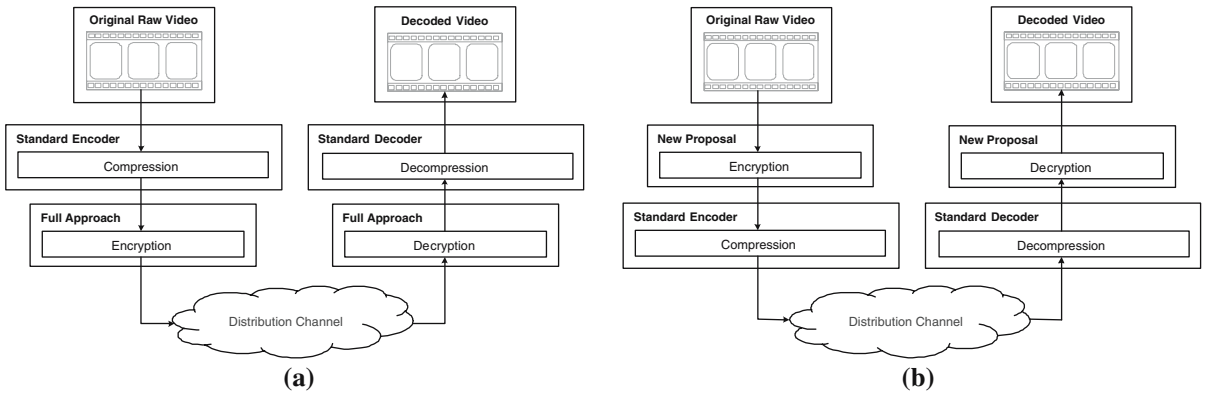

Fig. 3 The architecture for: **a** full video encryption algorithms proposed in the past almost all of which are post-compression approaches, and **b** the proposed method which is pre-compression approach resulting in no modification to the codec and fully application compliant video output

from the compressed bitstream (as shown in Fig. 2b), are format-defiant.

A notable path that the research community took for the full non-conventional video encryption approaches was the use of fast chaotic maps to achieve encryption. Chaos-based methods are apparently promising due to their fast performance. Although many chaotic encryption approaches were shown to be insecure, there are chaotic encryption algorithms that, up to date, remain unbroken (e.g. [16]). A good overview of these approaches, along with their comparative security analysis is presented in [11] and [7]. There are a few recently proposed fast, hardware-friendly, full encryption methods that are based on a class of neural networks [4,9]. However, these methods were later shown to be less secure than originally anticipated [5,23]. There are also non-conventional full approaches based on other mathematically hard problems, but most of them have been shown insecure due to oversimplification. For example, Yi et al. [31] proposed a new fast encryption algorithm for multimedia (FEA-M), which bases the security on the complexity of solving nonlinear Boolean equations. The scheme was shown insecure against several different attacks [18,19,29,30]. Later its improved version from [18] was shown insecure against differential chosen-plaintext attack in [13].

In addition to questionable security, the full non-conventional encryption approaches are, as noted earlier, not application-friendly when applied after compression and application requirements such as format-compliance or error-resilience are not supported. Figure 3a shows the typical structural design of full encryption approaches, where encryption is performed after compression and where no modification to the codec is required.

With the adoption of AES in 2001, which is a much faster and more secure cryptosystem than DES, application issues concerning only the encryption processing speed are less significant. In light of this, most fast full non-conventional approaches proposed in the past are, to a certain degree, obsolete.

In this work we propose a video encryption mechanism that securely encrypts the video stream before compression

(pre-compression approach), but that also preserves data correlation present in a typical video sequence. The architecture of this kind of approach is shown in Fig. 3b. In particular, we present a model that uses the permutation-based transformations to achieve this goal.

4 The proposed model: correlation-preserving encryption for digital videos

Strong encryption is a process that produces randomized data. On the other hand, compression efficiency is directly dependent on the presence of source data redundancy. The more the data is correlated, the better the compression, and vice versa. One may ask the following important question: is it possible to design an encryption mechanism of reasonable security that preserves, or perhaps even increases the compressibility of data? Such system is here referred to as the *correlation-preserving encryption* (CPE). CPE for digital videos encoded with spatial-only coding is possible to achieve with permutation-based transformations.

4.1 Notation

Let $\mathcal{V}$ be a video sequence consisting of m frames denoted by $I_1, I_2, \ldots, I_m$. For our model we assume that all frames in $\mathcal{V}$ are part of a single scene with a relatively low movement, captured with a static camera, so that the differences between adjacent frames are relatively small. Furthermore, we assume that each frame has a dimension of $w \times h$ and up to 2^n different pixel values (colors). Finally, let σ_i denote a canonical sorting permutation of I_i, and $\sigma_i(I_i)$ the image with sorted pixels from I_i. For a given frame I there are a large number of sorting permutations for I. By a *canonical* sorting permutation of I we mean a *unique* sorting permutation σ that any two distant parties can compute solely by knowing I, which is the case when the parties utilize the same computational method. For example, the communicating parties can agree on always choosing the lexicographically smallest sorting permutation of frame I. A more efficient method for generating a canonical sorting permutation relies on using a standard sorting algorithm such as quicksort. A quicksort-based algorithm that was used in our experiments is presented in Sect. 6.

4.2 The algorithm: encryption and decryption

We describe two versions of our algorithm. The first one is designed for lossless spatial-only codecs, such as Animated GIF (A-GIF), Motion PNG (M-PNG), or Motion Lossless JPEG (M-JLS), while the second one is targeted for lossy spatial-only codecs, such as Motion JPEG (M-JPEG).

Suppose Alice wishes to securely transmit a video sequence $\mathcal{V} = I_1, I_2, \ldots, I_m$ to Bob. We assume that if Alice would like to transmit $\mathcal{V}$ to Bob non-securely, she would normally use video compression algorithm C (the encoder) and decompression algorithm D (the decoder). She opens two channels with Bob, the regular, non-secure multimedia distribution channel R, and a second, secure channel S where transmission data is encrypted using some standard method (e.g. AES-based protocol).

4.2.1 The lossless case

The following is the proposed encryption algorithm for lossless codecs (see Fig. 4):

1. Given a video sequence $\mathcal{V} = I_1, \ldots, I_m$, Alice computes σ_1.
2. Alice calculates $C(I_1)$ and transmits it through channel S. This is the secret part (the key) of the algorithm.
3. For each subsequent frame I_i, $i = 2, \ldots, m$, Alice does the following:
   (a) She computes the frame $\sigma_{i-1}(I_i)$ and the permutation σ_i;
   (b) Alice then applies the standard encoder to the frame $\sigma_{i-1}(I_i)$ and transmits the encoded frame $C(\sigma_{i-1}(I_i))$ to Bob via the regular, non-secure multimedia channel R.

At the other end, Bob performs the following decryption algorithm (see Fig. 5) in order to recover the original video sequence $\mathcal{V}$:

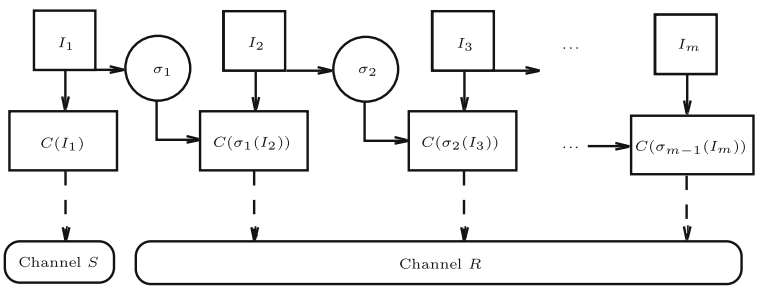

Fig. 4 Diagram of the proposed encryption algorithm for lossless spatial-only video codecs

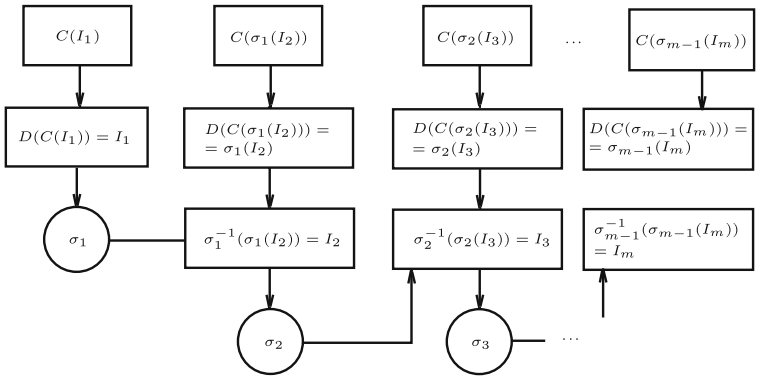

Fig. 5 Diagram of the proposed decryption algorithm for lossless spatial-only video codecs

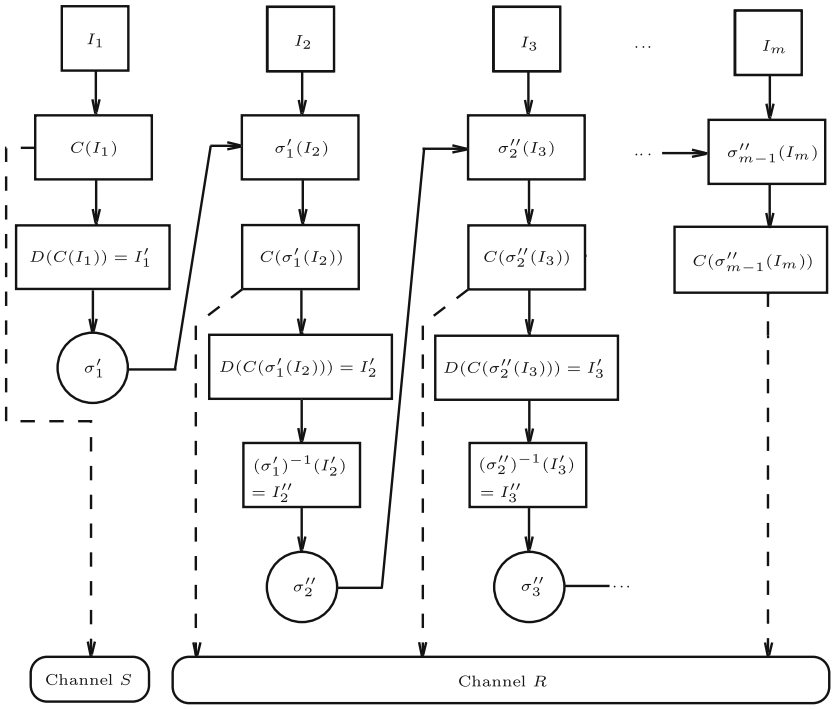

Fig. 6 Diagram of the proposed encryption algorithm for lossy spatial-only video codecs

1. Bob decodes $C(I_1)$ into I_1 and obtains a canonical sorting permutation σ_1.
2. For each received frame $C(\sigma_{i-1}(I_i))$, $i = 2, \ldots, m$, Bob does the following:
   (a) Decodes $C(\sigma_{i-1}(I_i))$ into $\sigma_{i-1}(I_i)$ and calculates $I_i = \sigma_{i-1}^{-1}(\sigma_{i-1}(I_i))$ where σ_{i-1}^{-1} is the inverse permutation of σ_{i-1};
   (b) Calculates the canonical sorting permutation σ_i of I_i.

4.2.2 The lossy case

The proposed encryption algorithm for lossy codecs (see Fig. 6) is as follows:

1. Given a video sequence $\mathcal{V} = I_1, \ldots, I_m$, Alice first computes $C(I_1)$ and then $I_1' = D(C(I_1))$ from which she obtains the canonical sorting permutation σ_1'.
2. Alice sends $C(I_1)$ via secure channel S to Bob.
3. She applies σ_1' to I_2, computes $C(\sigma_1'(I_2))$, and sends it via R to Bob.
4. Next, she computes $I_2' = D(C(\sigma_1'(I_2)))$ and then $I_2'' = (\sigma_1')^{-1}(I_2')$ from which she calculates the canonical sorting permutation σ_2''.
5. For each subsequent frame I_i, $i = 3, \ldots, m$, Alice does the following:
   (a) Applies σ_{i-1}'' to I_i, computes $C(\sigma_{i-1}''(I_i))$, and sends it to Bob via the regular channel R;
   (b) Computes $I_i' = D(C(\sigma_{i-1}''(I_i)))$;

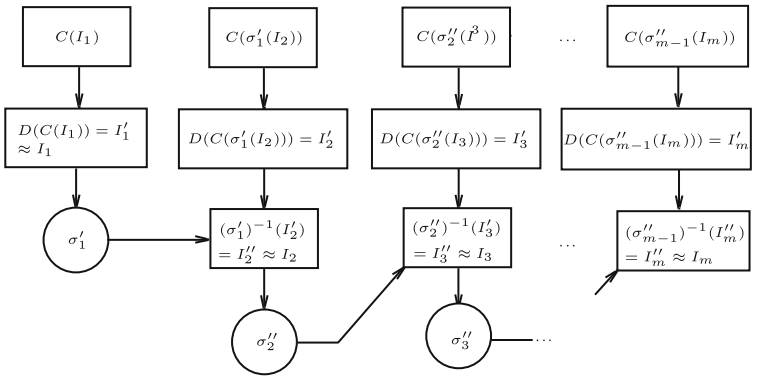

Fig. 7 Diagram of the proposed decryption algorithm for lossy spatial-only video codecs

(c) Applies $(\sigma''_{i-1})^{-1}$ to get $I''_i = (\sigma''_{i-1})^{-1}(I'_i)$;
(d) Calculates the canonical sorting permutation σ''_i.

At the receiver's side, Bob performs the following decryption algorithm (see Fig. 7) to recover the approximation of the original video sequence $\mathcal{V}$:

1. Bob calculates $D(C(I_1)) = I'_1 \approx I_1$ and sorting permutation σ'_1.
2. From $C(\sigma'_1(I_2))$ he computes $I'_2 = D(C(\sigma'_1(I_2)))$.
3. Bob approximates $I_2 \approx I''_2 = (\sigma'_1)^{-1}(I'_2)$.
4. He then recovers the canonical sorting permutation σ''_2 of I''_2.
5. For each received frame $C(\sigma''_{i-1}(I_i)), i = 3, \ldots, m$, Bob does the following:
   (a) Decodes $C(\sigma''_{i-1}(I_i))$ into $I'_i = D(C(\sigma''_{i-1}(I_i)))$;
   (b) Approximates $I_i \approx I''_i = (\sigma''_{i-1})^{-1}(I'_i)$;
   (c) If $i < m$ he calculates a sorting permutation σ''_i of I'_i.

The lossy version of our proposed algorithm requires a compression stage as a preprocessing to the encryption, so technically it does not exactly corresponds to the Fig. 3b. When compression is seen as a preprocessing step, the algorithm should still be considered to be a pre-compression approach, and as such, inherently possesses the nice properties such as codec-standard compliance and format-compliance.

4.3 Discussion

Intuitively, our permutation-based encryption approach works as follows. Sorted, as well as "almost sorted" frames are quite compressible, and in many instances even more compressible than the original source frames. When a sorting permutation of the previous frame acts on the current frame, it produces an almost sorted frame. In general, transmitting permutations for a frame to the receiver is an expensive transmission. However, transmitting a compressed frame from which the initial permutation can be computed is cheap. Once an initial permutation is transmitted via secure channel, the sender uses it to "almost sort" the next frame. In the next section we show that a sorted or almost sorted frame can be safely sent via regular channel. By calculating a sorting permutation of the received frame, the receiver uses it to recover the next frame, and so on. This way the spatial correlation within frames of a video sequence is expected to be preserved, if not improved, when static-camera low-motion sequences (e.g. video conferencing or telephony) and spatial-only video codecs (e.g. M-JPEG) are used.

5 Security analysis of our model

This section serves to analyze security aspects of the proposed method. The security strengths and weaknesses of the proposed system are pointed out.

Brute-force attack. Brute-force attack is based on exhaustive key search, and is feasible only for the cryptosystems with relatively small key space. In our case, the brute-force attack consists of two possible venues: one could either attack the underlying conventional cryptosystem used for encryption in channel S, or the proposed permutation-based method used in channel R. For that reason, it is recommended to use a strong conventional symmetric-key cryptosystem, such as AES with 128-bit or stronger keys. The size of the key space related to our permutation-based method is equivalent to the following: Given a color histogram of an $w \times h$ image I, how many different images can be formed out of the histogram color values? Note that I is just one of these images.

Let $U(I) \subset \mathbb{Z}_{2^n}$ denote the set of unique pixel values that appear in I. Furthermore, if $u \in U(I)$, let $N(u)$ denote the number of times pixel value u appears in I. By a relatively well-known counting result we have that the number of different images that can be formed out of pixels from I, where $U(I) = \{u_1, \ldots, u_k\}$, is

$$|S_{w\times h}(I)| = \frac{(w \times h)!}{\prod_{i=1}^{k} N(u_i)!}.$$

Thus, there are exactly $(w \times h)!/\prod_{i=1}^{k} N(u_i)!$ different images, with exactly the same frequency distribution of $\{N(u_i) : i = 1, \ldots, k\}$. These distinct images determine the effective key space of our method. Thus, if one uses an n-bit conventional cryptosystem to encrypt key frames in channel S, the actual key space of the proposed method is

$$\min\left(2^n, \frac{(w \times h)!}{\prod_{i=1}^{k} N(u_i)!}\right).$$

Clearly, the size of the key space depends on the color histogram of the encrypted frame. As one can see, this number is extremely large when considering any meaningful images of reasonable dimensions, and it is usually much larger than brute-forcing 2^n keys of the used conventional symmetric-key cryptosystem.

Known/chosen-plaintext and chosen-ciphertext attacks Permutation-only video encryption is considered weak against known/chosen-plain-text attack, and a chosen-ciphertext attack [14]. However, all of the previously proposed methods rely on generating the secret permutation using a secret key. Under this scenario, all of the aforementioned attacks are trying to recover the secret key (or a part of it) that was used for the current or future encryptions. Our scheme does not rely on such a principle, and there is no secret key upon which a permutation is generated. Our method relies on the sorting permutation of the *previous* frame, and thus, a key is directly dependant of the plaintext. Under a chosen-plaintext attack, the adversary can compute the sorting permutation for the chosen frame, but this gives no information about the sorting permutations for the unknown frames. Under a chosen-ciphertext attack, the adversary can recover the unsorting permutation for the chosen encrypted frame, but this gives no information regarding other unknown ciphertexts.

Known weaknesses A limited known-plaintext attack is applicable to our method, because the adversary can recover all frames that follow the known frame until the scene changes and key frame is updated. This, however, only reveals that one scene, since the key is completely changed as soon as the scene changes. This is a feature of all systems whose key depends on the plaintext. In addition, if the adversary has the information on the possible videos to be encrypted, he or she may be able to recognize which video sequence is being transmitted from Alice to Bob by observing the publicly given pixel value histograms of frames. Another related problem is the adversary's ability to analyze the properties of a given histogram for rough clues about the content. Namely, cartoon pictures and real photos have different histograms, and photos of human faces usually have narrower histograms than photos of natural scenes [11]. Although limited, these attacks are unavoidable in the proposed scheme and the scheme should not be used when conditions are such that these attacks are possible to launch by an adversary.

6 Implementation issues

There are a few implementation issues related to the proposed scheme. The compression performance of the proposed method can be further improved by knowing how the compression works at the encoder side. In particular, since M-JPEG performs a DCT-based compression on 8×8 blocks of a frame, better compression is achieved if the "almost sorted" data is reordered by filling in the 8×8 blocks in a frame instead of applying the usual raster ordering. This step does not impact the computational complexity of the proposed method and it notably improves the performance of M-JPEG encryption.

The second issue involves the calculation of a sorting permutation for a given frame. An efficient way of calculating such a permutation is by using a modification of a fast sorting algorithm, such as *quicksort* [10]. A given sorting algorithm should be modified so that, in addition to keeping track of the exchanged elements, it keeps track of the indices of the exchanged elements. Both the sender and the receiver should use the same sorting method in order to obtain the same sorting permutation.

The following recursive algorithm for obtaining a canonical sorting permutation (with zero-based index) of an image is used in our experiments. The algorithm takes four input variables, and the initial input is: (1) a copy a of a $w \times h$ image I, (2) $p = [0\ \ 1\ \ 2\ \ \ldots\ \ (w \times h) - 1]$ (the identity permutation with zero-based index), (3) $l = 0$ and (4) $r = (w \times h) - 1$

1. Set $i = l - 1$, $j = r$, and $v = a[r]$
2. If $r \leq l$ return from the algorithm
3. Start an infinite loop and do the following:
   (a) Set $i = i + 1$
   (b) While $a[i] < v$ do the following:
      i. Set $i = i + 1$
   (c) Set $j = j - 1$
   (d) While $v < a[j]$ do the following:
      i. If $j = l$ break from this while loop
      ii. Set $j = j - 1$
   (e) If $i \geq j$ break from the infinite loop
   (f) Exchange $a[i]$ and $a[j]$
   (g) Exchange $p[i]$ and $p[j]$
4. Exchange $a[i]$ and $a[r]$
5. Exchange $p[i]$ and $p[r]$

6. Recursively call this algorithm with $a = a, p = p, l = l$ and $r = i - 1$
7. Recursively call this algorithm with $a = a$, $p = p$, $l = i + 1$ and $r = r$

Finally, there is a need for having self-decodable frames, ones that are independent of previous or future frames. In the base scheme, the current frame is always recoverable from the sorting permutation of the previous frame, and as such, the scheme cannot handle VCR-like functionality or frame dropping caused by noisy channels or other communication errors. There is a simple and straightforward extension by which these functionalities can be achieved: the sorting permutation of the first frame (the key frame) can be used to "almost sort" every kth frame. The loss in compression gain is expected to be small since the assumption that all frames are part of a single scene holds. By doing so, the receiver can fast forward or rewind the video up to a kth frame, and frame dropping will affect only frames up to the next kth frame. This strategy is analogous to the strategy used in MPEG-like algorithms, where GOPs (group of pictures) with repetitive I-frames are utilized.

7 Performance analysis and experimental results

To evaluate the performance of our method in terms of compression, we run experiments on several fairly static grey-scale sequences in CIF and QCIF formats. As Tables 1 and 2 show, our method preserves, and in many instances even improves the compression performance of the original codec without encryption. Table 2 also compares the loss of quality (in terms of PSNR) when our method is applied to the lossy M-JPEG with quality parameter Q that controls the quantization level in M-JPEG. Q ranges from 0 (the worst quality) to 100 (the best quality). A modest loss of quality occurs by performing the proposed encryption with lossy codecs.

Figures 8 and 9 show in more detail how the compression ratios of A-GIF and M-JLS lossless codecs with and without the proposed method compare when applied to sequences *Akiyo* and *Mother Daughter*. As the figures depict, the compression results are better for *Akiyo* since it is a more static sequence than *Mother Daughter* (see Fig. 11). Figure 10 shows the compression performance of a lossy codec (M-JPEG with quality parameter $Q = 90$) with and without the proposed method applied to the same two sequences. Again the results are better for a more static of the two sequences.

Observe that the curves from Figs. 8, 9, and 10 corresponding to our approach are, for the most part, similar amongst themselves and also to the corresponding curves in Fig. 11, which depicts the *mean square errors* (MSEs) between the consecutive frames within *Akiyo* and *Mother Daughter* sequences. In essence, this phenomenon occurs due to the type of transformations we apply. This observation indicates that as long as the rate of change between the consecutive

Table 1 Performance of our method for lossless spatial-only codecs

		Codec w/o encryption		Codec w/ encryption	
Sequence	Codec	Size (MB)	Avg. Fr. (KB)	Size (MB)	Avg. Fr. (KB)
Akiyo	A-GIF	21.53	73.51	9.24	31.54
CIF (300 frms)	M-PNG	19.09	65.15	7.96	27.17
	M-JLS	10.70	36.53	7.39	25.23
Mother and	A-GIF	21.68	73.99	15.93	54.38
Daughter	M-PNG	19.23	65.64	14.97	51.11
CIF (300 frms)	M-JLS	11.31	38.62	12.63	43.12
Monitor Hall	A-GIF	24.88	84.91	18.50	63.14
CIF (300 frms)	M-PNG	21.67	73.97	17.55	59.91
	M-JLS	13.13	44.83	14.62	49.89
Grandma	A-GIF	2.06	21.91	1.45	14.84
QCIF (100 frms)	M-PNG	1.90	19.46	1.30	13.34
	M-JLS	1.18	12.13	0.89	9.09
Claire	A-GIF	1.52	15.59	1.17	11.98
QCIF (100 frms)	M-PNG	1.35	13.86	1.02	10.48
	M-JLS	0.75	7.68	0.71	7.27
Miss	A-GIF	1.54	15.80	1.33	13.57
America	M-PNG	1.44	14.79	1.27	13.05
QCIF (100 frms)	M-JLS	0.81	8.34	0.91	9.37

Table 2 Performance of our method for lossy spatial-only codecs

Sequence	Q	M-JPEG w/o encryption			M-JPEG w/ encryption		
		Size (MB)	Avg. Fr. (KB)	PSNR	Size (MB)	Avg. Fr. (KB)	PSNR
Akiyo	90	3.96	15.93	45.29	3.71	12.72	41.63
CIF (300 frms)	70	2.05	8.24	40.39	1.92	6.59	36.13
	50	1.55	6.24	38.20	1.37	4.70	33.93
Mother and	90	4.21	16.96	45.25	4.98	17.07	39.97
Daughter	70	2.24	9.04	40.91	2.48	8.48	34.68
CIF (300 frms)	50	1.70	6.84	38.88	1.72	5.91	32.70
Monitor Hall	90	5.55	22.35	43.21	6.05	20.71	38.89
CIF (300 frms)	70	3.02	12.15	38.12	2.93	10.03	33.49
	50	2.25	9.05	35.95	2.01	6.89	31.40
Grandma	90	0.57	5.94	41.04	0.35	3.58	41.55
QCIF (100 frms)	70	0.32	3.28	36.47	0.19	1.93	36.35
	50	0.24	2.48	34.81	0.14	1.44	34.06
Claire	90	0.41	4.22	45.19	0.31	3.19	42.23
QCIF (100 frms)	70	0.25	2.62	39.86	0.17	1.80	36.60
	50	0.20	2.08	37.51	0.13	1.36	34.36
Miss	90	0.35	3.64	45.63	0.36	3.69	41.55
America	70	0.19	1.98	41.52	0.19	1.98	35.97
QCIF (100 frms)	50	0.15	1.56	39.71	0.14	1.46	33.83

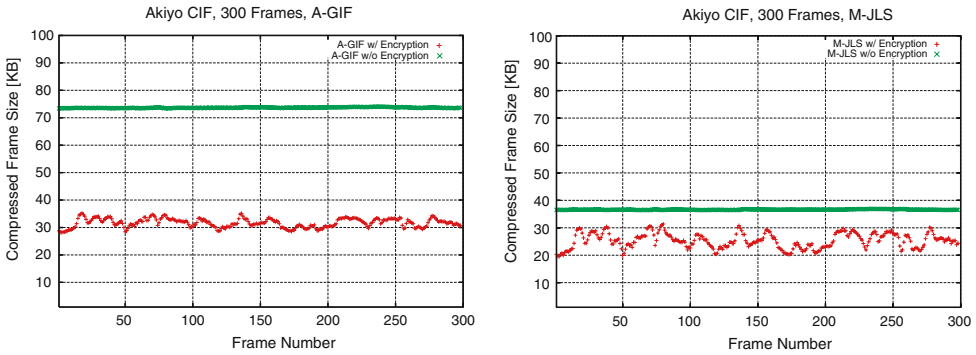

Fig. 8 Comparison of the compression performance for A-GIF (*left*) and M-JLS (*right*) codecs on **Akiyo** sequence with and without encryption

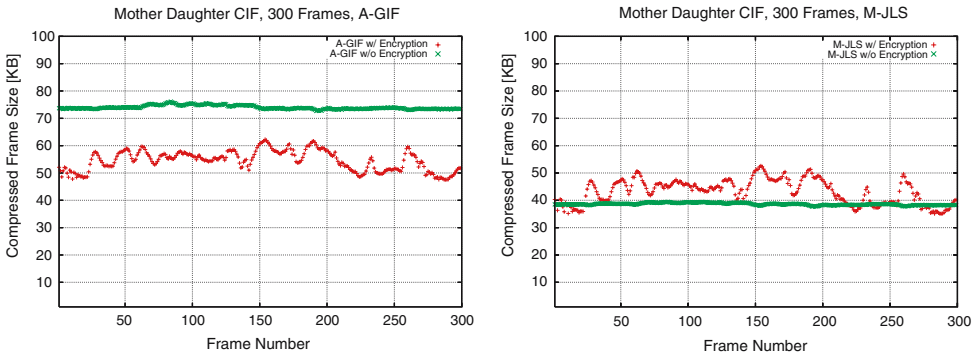

Fig. 9 Comparison of the compression performance for A-GIF (*left*) and M-JLS (*right*) codecs applied to *Mother Daughter* sequence with and without encryption

frames is small, the performance of our approach should be satisfactory. In Fig. 12, one can visually observe the appearance of the encrypted frames and the quality loss due to lossy encryption. There is a slight salt-and-pepper noise added to the reconstructed (decrypted) video as can be seen in Fig. 12f.

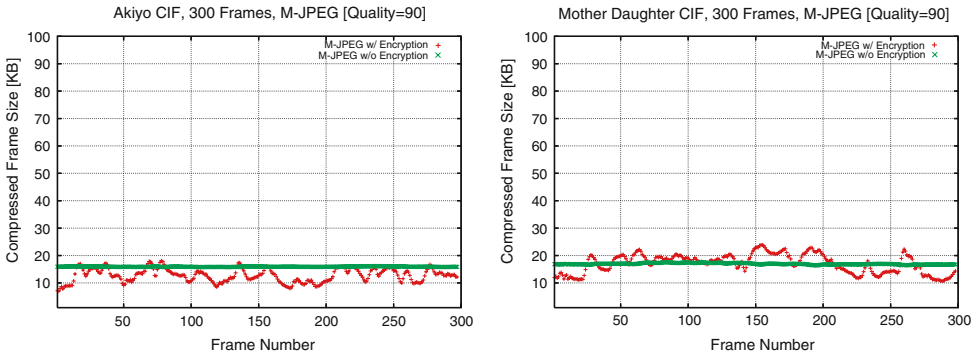

Fig. 10 Comparison of the compression performance for M-JPEG codec applied to *Akiyo* (*left*) and *Mother Daughter* (*right*) sequences with and without encryption

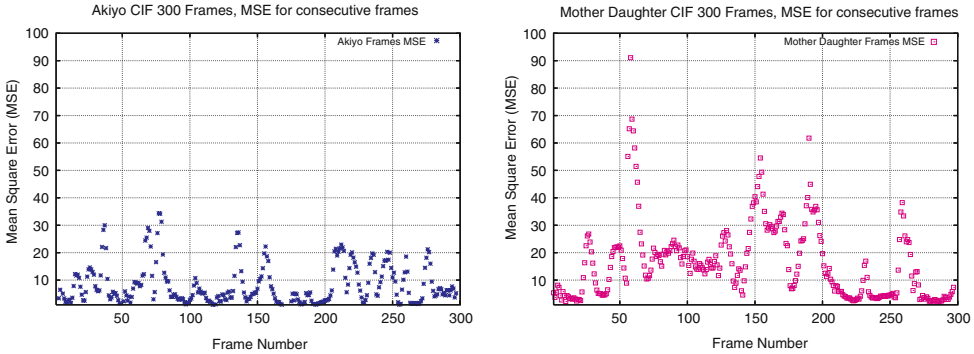

Fig. 11 Mean square error (*MSE*) between the consecutive frames of *Akiyo* (*left*) and *Mother Daughter* (*right*) sequences

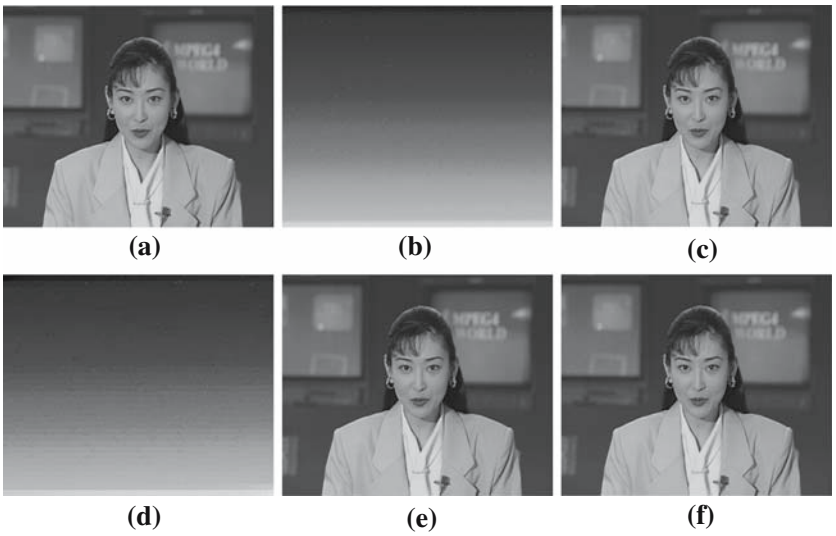

Fig. 12 The frame #150 of: **a** the original Akiyo (uncompressed) sequence, **b** the sequence obtained by encrypting Akiyo sequence with the proposed method for a lossless codec and decoding it without decryption, **c** the sequence obtained by properly decrypting an encrypted Akiyo sequence using the proposed method with a lossless codec, **d** the sequence obtained by encrypting Akiyo sequence with the proposed method for lossy M-JPEG codec (quality parameter $Q = 90$) and decoding it without decryption, **e** the sequence decoded from a regular, not encrypted encoded Akiyo sequence (compressed size: 16 KB, PSNR: 45.198), and **f** the sequence obtained by properly decrypting an encrypted Akiyo sequence using the proposed method with M-JPEG codec (compressed size: 12 KB, PSNR: 41.737)

Finally, we note that the computational complexity of the proposed method is very low at the decoder side for both lossless and lossy codecs, since the only additional computation that has to be performed involves the calculation of a sorting permutation. A standard sorting algorithm with complexity of $\mathcal{O}(N \log N)$ can be used to calculate a sorting permutation of the given frame. Inverting and/or applying a permutation is equivalent to a table lookup.

8 New steganographic architecture

The traditional digital video steganography is based on some form of data embedding. Permutation-based transformations allow for a different type of digital video steganography. The basic idea can be described as follows.

Let I and J be two natural images of the same size, somewhat similar histograms, and possibly completely different content. Furthermore, let σ_I and σ_J denote sorting, or "almost sorting" permutations of I and J, respectively. Then,

$$I' = \sigma_I^{-1}(\sigma_J(J)) = \sigma_I^{-1} \cdot \sigma_J(J) \quad \text{and}$$
$$J' = \sigma_J^{-1}(\sigma_I(I)) = \sigma_J^{-1} \cdot \sigma_I(I),$$

where $\cdot$ denotes the product of two permutations. The images I' and J' appear, at least to a human visual system, similar to images I and J, respectively. Figure 13 shows an example of this transformation: an image of Marilyn Monroe is transformed into an image of John Wayne, and vice versa.

These transformations can be used to realize a steganographic scheme within the framework of our proposed video encryption approach. Namely, instead of encrypting the source video $\mathcal{V}$ into "almost sorted" frames, given a different video sequence $\mathcal{W}$, the sender can further transform the "almost sorted" frames according to the inverse sorting permutation of the frames of $\mathcal{W}$. At the receiving end, the receiver easily recovers the "almost sorted" frames by knowing the actual sequence $\mathcal{W}$. Here, it is assumed that the sender and the receiver previously agreed on the hiding video sequence $\mathcal{W}$. For example, $\mathcal{W}$ could be some publicly known sequence. Once the "almost sorted" frames are obtained, our method resumes as previously proposed. Figure 14 shows the output of the proposed steganographic method where a sequence *Grandma* is disguised with a sequence *Claire*.

9 Conclusions and further research

In this work, along with providing an extended comprehensive classification of video encryption algorithms, we propose a novel video encryption algorithm designed for both lossless and lossy low-motion spatial-only video codecs. The algorithm preserves, and in many instances even improves the spatial correlation of the source data. The proposed encryption method can thus be performed before compression at the encoder side, and after decompression at the decoder side, a unique and often desirable feature. In effect, the algorithm produces fully application-friendly output, and requires no modification to the codec modules. We present both security and performance analysis of our method, and show that the algorithm is computationally efficient and resistant to typical cryptanalytic attacks. Finally, we introduce a new type of steganography as an extension to our encryption approach. The proposed steganographic scheme enables disguising a video with another video, which is a new concept in digital video steganography.

(a) **(b)** **(c)** **(d)**

Fig. 13 Permutation-based transformations: **a** the original image of Marilyn Monroe, **b** the original image of John Wayne, **c** the image of transformed Marilyn Monroe into John Wayne, and **d** the image of transformed John Wayne into Marilyn Monroe

Fig. 14 A new kind of digital video steganography using permutation-based transformations: *Grandma* video sequence is transformed into a degraded but visible version of *Claire* video sequence. As a comparison, A-GIF of the original *Grandma* sequence is 2.06 MB, while A-GIF of its steganographic encryption into *Claire* is 1.90 MB

Future directions should include an investigation of extending or modifying this principle to achieve efficiency in exploiting temporal correlation as well, in order to achieve applicability to more advanced video codecs such as H.26x and MPEG-x.

References

1. Alattar, A.M., Al-Regib, G.I.: Evaluation of selective encryption techniques for secure transmission of mpeg video bit-streams. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '99), 30 May–2 June 1999, Orlando, FL, USA, vol. 4, pp. 340–343. IEEE Computer Society
2. Agi, I., Gong, L.: An empirical study of secure mpeg video transmission. In: Proceedings of the IEEE Symposium on Network and Distributed System Security (SNDSS '96), 22–23 February 1996, San Diego, CA, USA, pp. 137–144. IEEE Computer Society
3. Bhargava, B., Shi, C., Wang, S.-Y.: Mpeg video encryption algorithms. Multimed. Tools Appl. **24**(1), 57–79 (2004)
4. Chan, C.-K., Chan, C.-K., Lee, L.-P., Cheng, L.-M.: Encryption system based on neural network. In: Proceedings of the IFIP TC6/TC11 5th Joint Working Conference on Communications and Multimedua Security (CMS '01), Darmstadt, Germany, in Communications and Multimedia Security Issuses of the New Century pp. 117–122. Kluwer, Boston 21–22 May 2001
5. Ćulibrk, D., Socek, D., Sramka, M.: Cryptanalysis of the block cipher based on the hopfield neural network. In: Proceedings of the Fifth Central European Conference on Cryptology (MoraviaCrypt '05), 15–17 June 2005, Brno, Czech Republic, Tatra Mountains Mathematical Publications (2005, to appear)
6. Eskicioglu, A.M.: Protecting intellectual property in digital multimedia networks. IEEE Computer, Special Issue on Piracy and Privacy (2003), 39–45
7. Furht, B., Muharemagic, E.A., Socek, D.: Multimedia security: encryption and watermarking. Multimedia Systems and Applications, vol. 28, Springer, December 2005
8. Furht, B., Socek, D., Eskicioglu, A.M.: Multimedia security handbook, Internet and Communications Series, vol. 4, chap. Fundamentals of Multimedia Encryption Techniques pp. 95–132. CRC Press (2004)
9. Guo, D., Cheng, L.-M., Cheng, L.L.: A new symmetric probabilistic encryption scheme based on chaotic attractors of neural networks. Appl. Intelli. **10**, 71–84 (1999)
10. Knuth, D.E.: The art of computer programming, 2nd edn., vol. 3: Sorting and Searching, pp. 113–122. Addison–Wesley, Reading, MA (1998)
11. Li, S., Chen, G., Zheng, X.: Multimedia security handbook. Internet and Communications Series, vol. 4, chap. Chaos-Based Encryption for Digital Images and Videos, pp. 133–167. CRC Press, West Palm Beach (2004)
12. Liu, X., Eskicioglu, A.M.: Selective encryption of multimedia content in distribution networks: Challenges and new directions. In: Proceedings of the Second IASTED International Conference on Communications, Internet and Information Technology (CIIT 2003), pp. 527–533. Scottsdale, AZ, USA, IASTED, 17–19 November 2003
13. Li, S., Lo, K.-T.: Security problems with improper implementations of improved fea-m. J. Syst. Softw. (2006). http://dx.doi.org/10.1016/j.jss.2006.05.002
14. Li, S., Li, C., Chen, G., Bourbakis, N.G.: A general cryptanalysis of permutation-only multimedia encryption algorithms, IACR's Cryptology ePrint Archive: Report 2004/374, 2004
15. Lookabaugh, T.D., Sicker, D.C., Keaton, D.M., Guo, W.Y., Vedula, I.: Security analysis of selectively encrypted mpeg-2 streams. Multimedia Systems and Applications VI, Proceedings of SPIE, Orlando, FL, USA, vol. 5241 pp. 10–21. SPIE Publishers, September 7–11 2003
16. Li, S., Zheng, X., Mou, X., Cai, Y.: Chaotic encryption scheme for real-time digital video. In: Real-Time Imaging VI, Proceedings of SPIE, vol. 4666, pp. 149–160. SPIE Publishers (2002)
17. Meyer, J., Gadegast, F.: Security mechanisms for multimedia data with the example mpeg-1 video, Project description of SECMPEG, Technical University of Berlin. http://www.gadegast.de/frank/doc/secmeng.pdf
18. Mihaljević, M.J.: On vulnerabilities and improvements of fast encryption algorithm for multimedia fea-m. IEEE Trans. Consum. Eletron. **49**(4), 1199–1207 (2003)
19. Mihaljević, M.J., Kohno, R.: Cryptanalysis of fast encryption algorithm for multimedia fea-m. IEEE Commun. Lett. **6**(9), 382–384 (2002)
20. Qiao, L., Nahrstedt, K.: A new algorithm for mpeg video encryption. In: Proceedings of the First International Conference on Imaging Science, Systems and Technology (CISST '97), pp. 21–29. Las Vegas CSREA Press, 30 June–3 July 1997
21. Qiao, L., Nahrstedt, K., Tam, M.-C.: Is mpeg encryption by using random list instead of zigzag order secure? In: Proceedings of the IEEE International Symposium on Consumer Electronics pp. 226–229 2–4 December 1997. IEEE Computer Society, Singapore
22. Shi, C., Bhargava, B.: A fast mpeg video encryption algorithm. In: Proceedings of the Sixth ACM International Multimedia Conference, Bristol, UK, ACM Electronic Proceedings, 12–16 September 1998. http://turing.acm.org/sigs/sigmm/MM98/electronic_proc eedings/shi/
23. Socek, D., Ćulibrk, D.: On the security of a clipped hopfield neural network cryptosystem. In: Proceedings of the ACM Multimedia and Security Workshop (ACM-MM-Sec'05), pp. 71–75. ACM Press, New York City 2005, 1–2 August 2005
24. Spanos, G.A., Maples, T.B.: Performance study of a selective encryption scheme for the security of networked, real-time video. In: Proceedings of the 4th International Conference on Computer Communications and networks (ICCCN '95), pp. 2–10. IEEE Press, Las Vegas 20–23 September 1995
25. Seidel, T.E., Socek, D., Sramka, M.: Cryptanalysis of video encryption algorithms. Tatra Mt. Math. Publ. **29**, 1–9 (2004)
26. Shi, C., Wang, S.-Y., Bhargava, B.: Mpeg video encryption in real-time using secret key cryptography. In: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '99), Las Vegas, NV, USA, vol. 6, pp. 2822–2828. CSREA Press, 28 June– 1 July 1999
27. Tang, L.: Methods for encrypting and decrypting mpeg video data efficiently. In: Proceedings of the Fourth ACM International Multimedia Conference, pp. 219–230. Boston, ACM Press 18–22 November 1996
28. Uhl, A., Pommer, A.: Image and video encryption: from digital rights management to secured personal communication. In: Advances in Information Security, vol. 15. Springer, New York (2005)
29. Wu, H., Bao, F., Deng, R.H.: An efficient known plaintext attack on fea-m. In: Proceedings of the Fifth International Conference on Information and Communications Security (ICICS 2003), pp. 84–87. Huhehaote, China, vol. 2836. LNCS, Springer, Berlin, Germany 10–13 October 2003
30. Youssef, A.M., Tavares, S.E.: Comments on the security of fast encryption algorithm for multimedia (fea-m). IEEE Trans. Consum. Eletron. **49**(1), 168–170 (2003)
31. Yi, X., Tan, C.H., Siew, C.K., Syed, M.R.: Fast encryption for multimedia. IEEE Trans. Consumer Eletronics **47**(1), 101–107 (2001)