EasyChair Preprint

№ 9726

Video Cryptography with Chaos

Shreyashree Dasgupta, Pranjali Kodhe, Sichika Lokhande,
Sakshi Khiradkar and Shruti Kamble

February 16, 2023

Video Cryptography with Chaos

Project phase-I report submitted in partial fulfillment of requirement for the award of degree of

Bachelor of Engineering in Information Technology (IT)

By

Ms. Shreyashree Dasgupta

Ms. Pranjali Kodhe

Ms. Sichika Lokhande

Ms. Sakshi Khiradkar

Ms. Shruti Kamble

Guide

Prof. Sonali Guhe

Department of Information Technology

G H Raisoni College of Engineering, Nagpur

(An Autonomous Institute affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur)
Accredited by NAAC with "A+" Grade
Ranked 130th by NIRF, MHRD in the Engineering Category for India Ranking 2021,
Ranked 2nd by ARIIA 2020, MHRD in Private or Self Finance Institutions, 5 Star Rating by MIC, MHRD 2021

Nov. 2021

Video Cryptography with Chaos

Abstract:

Video encryption and decryption is something which is lesser known to the common people but a popular topic of research. The collective process of encryption and decryption of data is termed as Cryptography. Cryptography is that the science of knowledge security that has become a very awfully essential side of recent computing systems towards secured data transmission and storage. The exchange of digital knowledge in cryptography leads to totally different algorithms which will be classified into two cryptographic mechanisms: symmetric key in which the identical keys are used for encryption and decryption and asymmetric key in which uneven keys are used for encryption and decryption.

Images and videos are broadly used in numerous processes. As a result, the safety of image data from unauthorized access is crucial at the hands of the user. Image encryption plays a big role within the field of data hiding. Image hiding or encryption methods and algorithms range from simple spatial domain methods to more complicated and reliable frequency domain. Image Encryption using Rubik's Cube primarily based algorithmic method to remodel the image firmly so that no unauthorized user is able to decrypt the image. Image encryption have applications in several fields together with the internet communication, transmission, medical imaging etc.

First, in order to scramble the pixels of grey-scale original image, the principle of Rubik's cube is deployed which only changes the position of the pixels. Considering two random secret keys, the bitwise XOR is applied into the rows and columns. These steps can be recurrent till the number of iterations isn't reached. Numerical simulation has been performed to check the validity and also the security of the projected encryption algorithm. Same is implemented on videos also, for more security assurance. As of now there exists image cryptography using chaos only. Our service is unique since video cryptography using chaotic functions does not exist yet. Also, our service is to bring video encryption and decryption to the mainstream.

---

1. Introduction

Technology advancements have increased the popularity of multimedia applications such as video conferencing, digital TV, DVDs, Internet telephony and surveillance videos. Video transmission via public transmission channels is at risk of eavesdropping. Therefore, a trusty and real-time security management system is required to safeguard the keep and transferred video streams.

The high bit rates needed for sending digital videos create their transmission via network channels a vast challenge. Two types of redundancies inside the video information are removed by using video compression. The first type is spatial redundancy among pixels inside a frame, whereas the second is between serial frames. There are many temporal compression techniques; block-based motion compensation is one of them. This method is based on finding similar blocks between any two frames freelance of their location. However, this sort of motion estimation technique is long-lasting. Distance diluted search, locality-based search, and multidimensional search space methods are techniques used to speed up the search for motion estimation; though, all of them suffer from some loss within the resulting quality

Another challenge is securing video streams. The most straightforward video encryption algorithm (VEA) is to encipher the compressed video frames using one of the classical private key encryption schemes such as the Advanced Encryption Standard (AES).

However, it doesn't meet real time requirements for multimedia system applications. Thus, specific encryption algorithms are tailored to encrypt video streams.

Video encoding and compression are combined in three alternative ways depending on when the encryption function is applied. The encryption function is also applied before, or within, or once the compression function. If the encryption function is embedded inside the compression operation, the algorithm is claimed to belong to the joint-compression and encryption algorithms category. On the other hand, if it is applied before or once compression is done, the algorithmic rule then belongs to a compression independent class of algorithms.

The zig-zag permutation algorithm applies the encryption function within the compression function, more specifically, after the quantization step [9]. The 64 quantized DCT (Discrete Cosine Transform) coefficients for an eight × 8 pixels block are shuffled based on a random permutation list (secret key). Data Encryption Standard helps to encrypt the DC coefficients [9]. In VEA, a secret key is used to randomly modify the sign bits of the DCT coefficients of I-frames only [10]. On the other hand, to enhance the safety within the changed VEA, the sign bits of the motion vectors within the B-frames and P-frames also are encrypted. Further security enhancements are enclosed in real time VEA supported using conventional private key encryption scheme.

Random Data Encryption Algorithmic program could be selective VEA [13]. It selects a proportion of the I-blocks in I-, P- and B-frames for encryption which supports a pseudo-random sequence whose initial seed as the secret key. With a comparatively little encryption ratio, it achieves a comparable security level as that of the naïve algorithmic program. It is supposed that the video to be encrypted is coded using H.264 standard.

Wu and Kuo presented conferred a unique approach for integrating encryption with multimedia compression that turns entropy coders into encryption ciphers with several statistical models. They applied their approach with success to the Huffman coder and QM coder.

Socek present an encryption algorithm based on permutations that preserve spatial correlation. Their algorithm will be applied before the compression stage of a spatial only video encoder. They examine several modes of operating their scheme to meet the requirements of different applications.

Liu and Koenig proposed a compression-independent video encryption scheme that supported shuffling the blocks of a video frame then obscuring the shuffled video information by encrypting a small portion of this data using a conventional encryption algorithm. The rest of the divided video data is encrypted by XORing every piece with its predecessor.

Malladar and Sanjeev Kunte employed chaotic maps in their selective encryption algorithm in [17] suited to video on demand application. The I-frames are encoded by comparing the entropy values of the macro-blocks with a threshold value. If the entropy value of a macro-block exceeds the threshold value, then it is XORed with a keystream generated with the help of a chaotic map, whose initial value is the secret key of the encryption scheme.

Today, most of the information on the Internet is in the form of images, which may contain confidential information, such as patients' medical records. In daily life, clinics or public health centers sometimes find it challenging to determine patients' diseases. They need the help of hospitals or more experienced medical experts for analysis and diagnosis. Therefore, images of patients' medical records must be sent from the clinic/public health center to the destination hospital. The problem is patients' medical records are confidential and contains sensitive data. There are also regulations and legal protection of medical records. For example, Indonesia's medical records regulation can be seen in [1]. So there has to be a way to maintain the security of patients' medical records, which can be done by performing image encryption [2].

Basically, image encryption is a technique that is performed with the aim of protecting the content conveyed therein. This encryption is done by transforming the image into another form so that it does not contain meaningful information and cannot be understood visually or statistically. In general, there are two types of image encryption: traditional encryption and chaos-based encryption. Traditional encryption uses common

encryption algorithms, such as the Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), or Advanced Encryption Standard (AES). Chaos-based encryption uses a sequence of (pseudo-)random numbers called chaotic maps as a key for encrypting images. Of these two types, chaos-based encryption is more suitable for use with images because the image consists of information (i.e., image pixels) with high redundancy and correlation, and the resulting encrypted image will be random and have low correlation between pixels. In contrast, traditionally encrypted images have more patterns, so they are more vulnerable to attack. In this study, two chaotic map models will be used: Arnold's cat map for confusion or shuffling pixel positions and the Henon map for diffusion or changing gray values .

To generate a chaotic map, there are several parameters that can be described as the key to the encryption and decryption process in this system. The generated chaotic map is influenced by these values, so the encryption and decryption process must use exactly the same parameter values. The slightest change to this value will result in a different output matrix, so there must be a mechanism that ensures the encryption and decryption process uses the same parameter values. In , author use shared-key cryptography with Diffie-Hellman method to secure the key exchange process.

2. Background

In this section, the important tools used in our planed schemes are introduced.

2.1 YCbCr colour space

Color image displays are consumed by red, green and blue voltage signals. However, there is corelative repetition between these signals. This repetition causes wasting storage and transmission bandwidth needed. YCbCr is an alternate color image representation that has been successfully applied in video and digital photography systems [19].

YCbCr representation divides out a luma signal(Y) or light intensity signal to be hold on with high resolution and two chroma components (Cb blue-difference and Cr red-difference) that can be sub-sampled, compressed or otherwise treated independently for improved system potency.

YCbCr is an useful way of encoding color information, where the primary colors corresponding roughly to Red, Green and Blue are processed into perceptually meaningful information.

The transformation of an image from the RGB color model to the YCbCr color space is a linear transformation, that makes its computation quick and appropriate for compression utilized by televisions [20]

$$Y = 16 + 65.481\ R + 128.553\ G + 24.966\ B \quad (1)$$

$$Cr = 128 - 37.797\ R - 74.203\ G + 112.0\ B \quad (2)$$

$$Cb = 128 + 112.0\ R - 93.786\ G - 18.214\ B \quad (3)$$

2.2 Chaotic maps

Chaotic maps have been extensively used in cryptography literature for their capability of generating random-looking sequences that are highly sensitive to the choice of the initial seed point [21, 22]. Several chaotic maps are examined in this paper to develop the two proposed VEAs.

2.2.1 Arnold's cat map

The Arnold cat map is a two-dimensional (2D) invertible map that convert a point (x, y) to another point (x′, y′) using the following linear transformation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod\ (N) \quad (4)$$

where p and q are positive integers and N is the range of rows (or columns) of the square matrix whose elements are to be mixed up. The fact that the determinant of the transformation matrix is unity clarifies the inversion operation. When employed for image or video frame encryption, the Arnold cat map is used several times to increase security. In this map, after a definite number of times of applying it, the original image re-appears .

ACM is a chaotic map model that is used to randomize the pixel positions in an image. It was first introduced by Vladimir Arnold as a way to shuffle an image of a cat. Mathematically, this concept works by stretching and distorting a square shape and then reassembling it into the same shape .

Since it was introduced, ACM has been intended to randomize the pixel position of an image so that it does not look the same, which is a confusion technique. ACM works by scrambling a pixel's position without changing the value of the pixel itself. This can be done using the following formula [8,9]:

$$\begin{bmatrix} x_{n\prime} \\ y_{n\prime} \end{bmatrix} = A \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \quad (1)$$

$$\begin{bmatrix} x_{n\prime} \\ y_{n\prime} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod\ N \quad (2)$$

Although ACM is a chaotic map, if iterations are repeated many times, it is possible that the original image will be rearranged because the ACM concept relies on position randomization only. According to researchers, up to 3N iterations may be needed to return to the original image, where N is the dimension of the image .

Arnold's cat map is commonly used for image encryption by shuffling the image pixels but actually it can be used to encrypt other form of multimedia data. In and , there are good examples of audio encryption using Arnold's cat map for securing voice communication. Arnold's cat map can also be used to watermark an image or video, which is useful for tamper detection. In and , there are some good examples of image watermarking, and for video watermarking using Arnold's cat map.

2.2.2 *Baker's chaotic map*

The Baker chaotic map is another 2D chaotic map that has been efficiently applied to image encryption [23]. It randomly generate a square matrix of dimension N × N by shuffling the pixel positions based on a secret key. Baker's map is used to speed up image encryption while maintaining a high degree of security.

It is defined as a bijection B(x, y) to allow for the existence of a unique inverse map on the receiver's side. The secret key includes a vector of k integers, [n1,..., nk], chosen such that each integer ni divides N and set Ni = n1 + ⋯ + ni.

The pixel located at (x, y) with Ni ≤ x < Ni + ni and 0 ≤ y < N is mapped to the location

$$B(x,y) = \left[\frac{N}{\mathrm{ni}}(x - \mathrm{Ni}) + y\ mod\ \frac{N}{\mathrm{ni}}, \frac{\mathrm{ni}}{N}\left(y - y\ mod\ \frac{N}{\mathrm{ni}}\right) + \mathrm{Ni}\right] \quad (5)$$

The effect of applying this map is to divide the image array into k vertical rectangles of height N and width ni. The pixels in each rectangle are repositioned in a row. Rectangles are examined from right to

left starting from upper rectangles then followed by lower ones. In each rectangle, the pixels are scanned from the bottom left corner towards upper pixels.

2.2.2a Logistic Map and Its Properties

LM can be considered an example of a chaotic system with a simple definition but rather complicated behavior [44]. LM is a one-dimensional map, so each iteration of the map generates one value, called an iterate. The computations of LM utilize one parameter

$r \in (0,4)$

and an initial value

$x0 \in (0,1)$

. Iterate values

$xn \in (0,1)$

are computed by (1):

$xn+1=r \cdot xn(1-xn)$,

(1)

where

$n \in \{1,2,3,\cdots,N\}$

is the sequential number of iterates and *N* is the total number of iterates.

2.2.3 *Hénon's chaotic map:*

Hénon's map [25] iterates the initial seed point (x0, y0) to yield a random-looking sequence according to (8) and (9)

$$x_{i+1} = 1 + y_i - ax_i^2 \quad (8)$$

$$y_{i+1} = bx_i \quad (9)$$

The map depends on two parameters a and b, with common values of $a = 1.4$ and $b = 0.3$ to yield a complex behavior with time. In our scheme, this map is used to generate a keystream which is involved in the substitution step of our second algorithm.

2.3 *Compression techniques*

Video compression relies on eliminating repeated information, thus, creating a file smaller without affecting its quality. Spatial redundancy happens because pixels are typically replicated (with minor changes) at intervals in a single frame of a video. Temporal redundancy occurs when consecutive frames of a video display images of the same scene. It is customary for the content of the scene to be fixed or to change slightly between successive frames.

MPEG-2 consist of compressed video data referred to as a video stream. The fundamental unit of the video stream is a 'Group of Pictures' made up of three types of frames: I-frames, P-frames, and B-frames. The 'I'-frames are often restructured without any reference to other frames. On average, the 'I'-frames will occur only once every 10–15 frames of motion pictures. This kind of frame contains information only about itself. 'P'-frames can only be estimated by reference to a previous I-frame or P-frame; it is not possible to reconstruct them without any data of other frame. The 'B'-frames are spoken as to as bi-directional frames because they can be recreated based on forward and backward predictions from the data presented in the closest preceding and following 'I' or 'P' frame.

MPEG-2 compression algorithm involves four basic steps: pre- processing, temporal prediction, motion compensation and quantization coding. Pre-processing eliminates redundancy in color representation through conversion to the YCbCr color space and using a suitable sampling and sub-sampling scheme. The DCT transforms these signals into frequency coefficients which holds the color and brightness information. These signals can then be contracted more easily. Temporal redundancy elimination between frames; called as motion estimation step, is the most time-consuming part of MPEG encoding [20].

On the other hand, Lempel–Ziv–Welch (LZW) compression algorithm is a lossless algorithm that exploits spatial redundancy to make variable-length codes for color combinations [26].

3. Performance measures for evaluating a video encryption scheme

Assessing the quality of encryption is required because it demonstrates the strength/weakness points in the system. In video encryption techniques, there are special tests performed to conclude the quality of the encrypted video [27]. The encrypted video frames should be random-looking and showing sensitivity to little modification in the decryption key, which means getting a corrupted frame when decrypted using the incorrect key.

3.1 *Statistical measures*

Statistical measures are used to assess the randomness within the encrypted video frames.

The correlation (ρ) between two vectors X and Y of adjacent pixels in each frame in the video is calculated as

$$cov(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{j=1}^{N} x_j\right)\left(y_i - \frac{1}{N}\sum_{j=1}^{N} y_i\right) \tag{10}$$

$$D(X) = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_j - \frac{1}{N}\sum_{j=1}^{N} x_j\right)^2 \tag{11}$$

$$\rho = \frac{cov(X,Y)}{\sqrt{D(X)\,D(Y)}} \tag{12}$$

3.2 *Similarity and differential measures*

The differential measures show how sensitive the encryption scheme is to slight changes within the plain frame and little modification in the decryption key.

3.2.1 *Mean absolute error (MAE):*

A successful video encryption scheme should produce frames that are quite distinct from the first original video frame before encryption. The average absolute change in intensities between the encrypted video and the source video is calculated as follows:

$$MAE = \frac{1}{N\times N}\sum_{i=1}^{N}\sum_{j=1}^{N}|P(i,j) - E(i,j)| \tag{13}$$

where P(i, j) and E(i, j) denote the colour intensities of pixel (i, j) in the plain video frame and encrypted frame, respectively.

3.2.2 *Mean squared error (MSE):*

The MSE between two video frames X and Y is outlined as

$$MAE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}[X(i,j) - Y(i,j)]^2}{n \times m} \quad (14)$$

Obviously, the largest similarity is achieved when the MSE approaches 0.

3.2.3 *Number of pixels change rate:*

The number of pixels modification rate (NPCR) is employed to measure the percentage of pixels that are different between two videoframes and is calculated as follows [27]:

$$D(i,j) = \begin{cases} 0, E_1(i,j) = E_2(i,j) \\ 1, E_1(i,j) \neq E_2(i,j) \end{cases} \quad (15)$$

$$NPCR = \frac{1}{N \times N}\sum_{i=1}^{N}\sum_{j=1}^{N} D(i,j) \times 100\% \quad (16)$$

where E1and E2 are the two video frames. This measure can be employed to assess how sensitive a video encryption algorithm is to slight variations in the plain video frame.

3.2.4 *Unified average changing intensity (UACI):*

The average intensity of absolute differences between two frames is decided according to the following equation [27]:

$$UACI = \frac{1}{N \times N}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{|E_1(i,j) - E_2(i,j)|}{225} \times 100\% \quad (17)$$

Again, E1 and E2 are two video frames. It can be used to assess the sensitivity to slight changes in the plain video or key changes.

3.2.5 *Reconstruction quality measure:*

The peak-signal-to- noise-ratio (PSNR) is employed as a measure to determine the reconstruction quality. PSNR is essentially identical to MSE. It is the logarithmic representation of MSE. It is calculated as given in (18), where the maximum value of a color intensity usually set to 255

$$PSNR = 10log_{10}\left(\frac{Max^2}{MSE}\right) \quad (18)$$

4. Security attack models

An effective video encryption scheme should eliminate the correlation among nearest pixels along the vertical, horizontal and diagonal directions.

Moreover, the color histogram of an encrypted video frame should be uniform and separate from that of the original one.

In assessing the safety of a video encryption scheme, its resistance to some attacks should be investigated. In what follows, the classical encoder attacks such as known-plaintext and chosen- plaintext attacks (CPAs) to find the secret parameters of the algorithm are reviewed. Additionally, integrity attacks such as the salt and pepper attack and cropping attack are outlined.

- Known-Plaintext Attack

The attacker is aware of at least one sample pair of both the plaintext and the ciphertext. In this case, the attacker could use this information to split the algorithm.

- Chosen-Plaintext Attack

The attacker can specify a plaintext of its own choice and feeds it into the encryption module to get the corresponding ciphertext. It then uses the result to see the encryption key.

- Additive Noise and Cropping Attacks

An attacker can check an encrypted video and can modify it. A successful video encryption scheme should allow the recipient to detect that the received video has been change during transmission. In the additive noise attack, random noise is attached to the intercepted video frames by the attacker. The salt and pepper noise is applied in our analysis of the safety of the projected schemes. On the other hand, in the cropping attack, lot of measures of the encrypted video frame are deleted or cropped [28].

5. Proposed video encryption schemes

In this section, the two proposed video encryption schemes are described and their security is analysed. The main theme of the proposed schemes is to be simple and flexible making them suitable for real-time applications while being able to provide an appropriate level of security. In our description, it is assumed that the input video file is not compressed, for example, a video file with AVI extension. If the video file is already compressed, then proceed with the remaining steps of the algorithms directly.

The communicating parties may share the keys over a public channel using a suitable key agreement protocol [29] or they can be hardwired on a tamper-resistant card.

5.1 *VCMAB video encryption algorithm*

The encryption key components are:

- The index (i) of a frame designated as a pre-selected frame in our algorithm description (Fi).
- The parameters p and q of the Arnold map, as well as the number of times of applying it (R) and the number of times for the original frame to reappear (C).
- A vector of k integers which divide (N), where N is the number of columns of the video frame.
- A 2-bit key component to indicate how the RGB colour channels are interchanged.

On the sender's side, first, a swapping operation of the RGB colour channels is applied to each frame. Next, the resulting file is compressed based on MPEG-2 scheme. The pre-selected compressed frame (Fi) is then scrambled based on the Arnold map. Next, Baker's map is used to shuffle the blocks of the compressed frames other than the frame Fi. Finally, a substitution step, in which the values of the blocks are replaced by the result of XORing them with the blocks of the encrypted pre-selected frame, is applied.

Baker's chaotic map is employed to scramble the blocks in each compressed frame to produce uncorrelated random-looking frames. Encrypting the pre-selected frame and swapping the RGB colour channels enlarge the keyspace and improve the security of the algorithm. The bitwise exclusive-OR operation is carried out between the frames resulting from Baker's chaotic map and the pre- selected frame to produce more uniform histograms for the encrypted frames.

5.1.1 VCMAB video encryption steps:

The steps performed on the sender's side to encrypt a video file are summarised below.

a. Read the video file to be stored as a group of images called frames.
b. Apply keyed RGB colour channel swapping.

c. Apply the MPEG-2 compression technique.
d. Apply the Arnold map iteratively to scramble the blocks of the pre-selected frame (Fi) according to the pre-shared key.
e. Encrypt the rest of the frames using Baker's chaotic map.
f. The frames resulting from the previous step are XORed with the encrypted pre-selected frame (Fi).

5.1.2 VCMAB video decryption steps:

On the receiver's side, the steps are applied in reverse order and are listed below.

a. Identify the encrypted pre-selected frame (Fi).
b. Perform a bitwise XORing operation between the enciphered pre-selected frame and the remaining frames of the received video file.
c. Apply the Arnold map (C – R) times to decrypt the pre- selected frame.
d. Apply the inverse of Baker's map to decrypt the rest of the video frames.
e. Apply the MPEG-2 decompression algorithm to restore the original video file.
f. Apply the inverse swapping map of the RGB colour channels.

5.2 *VCMTH video encryption algorithm*

Another proposal for a secure VEA is presented in this section, which begins again by swapping the RGB colour channels and then MPEG-2 compression is applied. This algorithm is more computationally efficient. It also involves two steps: a permutation step based on Tinkerbell map and a substitution step based on Hénon's chaotic map. Tinker bell map is used to shuffle the blocks of a compressed frame according to the rounded values of the 2D sequence produced by it.

Hénon's 2D chaotic map generates a pseudo-random sequence of real numbers (xn). These numbers are used to generate a keystream to encrypt a video frame according to the following steps:

$$e_n = x_n \times 1000 \tag{19}$$

$$f_n = e_n \tag{20}$$

$$k_n = f_n \bmod 256 \tag{21}$$

By XORing a byte of the plain frame (Pn) with the key (kn), the encrypted byte (Cn) can be obtained.

The encryption key components are:

- The four control parameters of the Tinker bell map as well as its initial seed 2D point.
- The two parameters of the Hénon map as well as its initial seed 2D point.
- A 2-bit key component to indicate how the RGB colour channels are interchanged.

It is noteworthy that if not enough bandwidth is available for communicating all these parameters, only the initial seeds of the two chaotic maps are kept secret. In this case, the control parameters of the two maps can be system-wide common parameters.

5.2.1 VCMTH video encryption steps:

On the transmitter's side, the following steps are applied to encrypt a video file.

a. Read the video file to be stored as a group of images.
b. Apply keyed RGB colour channel swapping.
c. Apply the MPEG-2 compression technique.
d. Tinkerbell chaotic map is applied to each frame to shuffle its components.

e. Hénon's map is used to generate the keystream according to (19)–(21).
f. The frames resulting from step (d) are XORed with the generated keystream.

5.2.2 VCMTH video decryption steps:

On the receiver's side, the recipient restores the original video by applying the following steps.

a. Hénon's map is used to generate the keystream according to (19)–(21).
b. The received frames are XORed with the keystream.
c. Apply the inverse Tinkerbell chaotic map to restore the blocks to their original positions.
d. Apply decompression.
e. Apply the inverse swapping scheme to the RGB colour channels to obtain the original video file.

5.3 *Key space analysis*

In what follows, we demonstrate that the keyspace is large enough to combat exhaustive search attacks mounted on any of our proposed schemes.

For our VCMAB VEA, the associated keyspace is large enough based on the following arguments:

- Assume the index of the pre-selected frame is restricted to the first w frames. Clearly, the larger the value of w, the security is enhanced.
- The Arnold map parameters p and q cannot exceed N as well as the parameter R providing the number of applications of the map.
- The parameters of Baker's map are restricted to permutations of a subset of size k of the divisors of N with repetitions of divisors being allowed. It is noteworthy that k itself is a component of the shared secret key to improve the security of the scheme.
- 2-bits for the RGB colour channels swapping scheme.

Now, let us consider the following numerical example. Let N equal 256. N has ∼1063 possible subsets of divisors, which is large enough to combat brute-force attacks against the security of the scheme. The whole keyspace size is O(2563 × 1063). The role of the substitution step is not restricted to enlarging the keyspace size, but it is essential to obtain a uniform histogram for the encrypted frames, which cannot be otherwise achieved.

The proposed VCMTH algorithm keyspace size is mainly dependent on the precision used in representing the Tinkerbell map parameters and initial seed. For precision of 10−10, the keyspace size is (1010)6 which is again large enough to withstand exhaustive search attacks. This is in addition to the initial seed and parameters of the Hénon map.

5.4 Resistance to known plaintext attack

Assume that the attacker knows the original video file P and the corresponding encrypted video file C. Regarding the VCMAB algorithm, before mounting any attack, it must essentially know the RGB colour channels swapping scheme to successfully decompress the encrypted video file and start any comparisons between the original and encrypted frames. The swapping scheme is obscured through employing the chaotic maps while encrypting the video frames. Similar arguments for our VCMTH scheme hold.

5.5 *Resistance to chosen plaintext attack*

Assume the attacker chooses all zero compressed video frames P and observes the encrypted video file C. Such a file cannot reveal any information about the VCMAB algorithm, and may reveal the keystream

generated by the Hénon map for the VCMTH algorithm. Consequently, such a video file should be rejected by the system decoder to achieve the desired security.

5.6 *Computational time complexity*

The three key steps in the first proposed encryption process are: iterative application of the Arnold map, pixels permutations of each frame in a video using Baker's chaotic map and bitwise XOR operation with a pre-selected encrypted frame. The Arnold map is applied R times to the pre-selected frame-blocks and therefore this step is O (N2). Baker's chaotic map for shuffling the blocks of each frame is also O(N2). Thus, the whole algorithm is O (cN2), where c is the number of frames of the compressed video file. Similar arguments can be made for the second proposed algorithm.

6. Simulation results and security assessment

We applied our proposed algorithms to a sample of standard videos including Foreman, Xylophone, Balcony and Aircrash. The performance of the proposed schemes is evaluated using the measures mentioned in Section 3.

6.1 *Visual testing*

As apparent in Figs. 1 and 2, the histograms of the encrypted frames using both algorithms are flat with all colour levels being equally likely. They are quite distinct from the histograms of the original frames. Moreover, both algorithms show great sensitivity to minor changes in the key as apparent in Figs. 3 and 4 as indicated by the high values for the differential measures. It is clear that decryption using a wrong key gives completely distorted frames. However, the VCMAB algorithm shows greater sensitivity to decryption under a slightly different key since it yields higher values for the differential measures.

The salt and pepper noise attack has been applied to some encrypted video frames with density 0.05 and the results are also shown in Figs. 3 and 4. The decrypted video frames are quite noisy and thus the receiver detects that the received video file has been

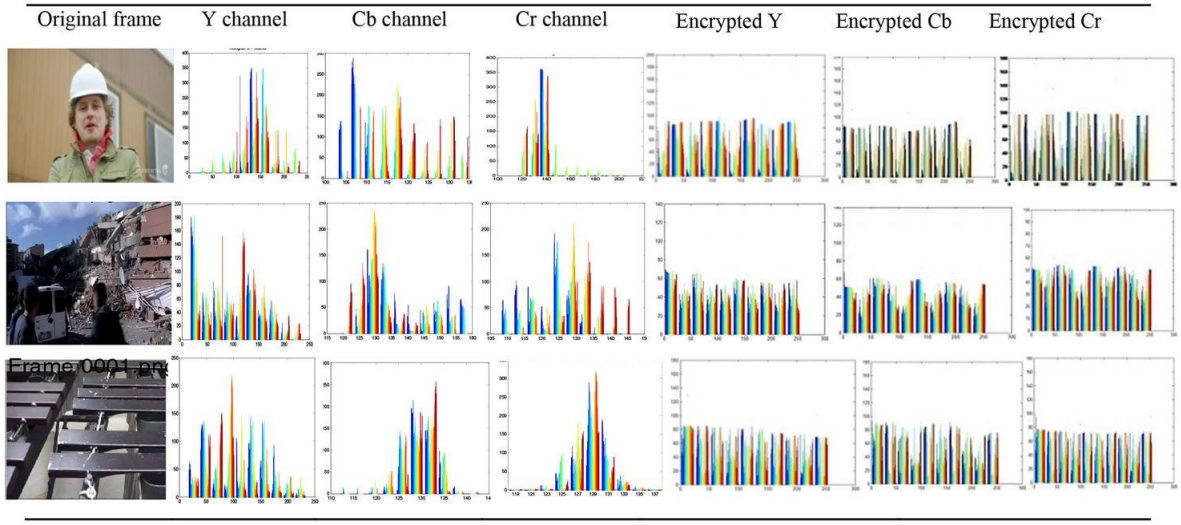

Fig. 1 *Histogram analysis of test videos before and after encryption using VCMAB algorithm*

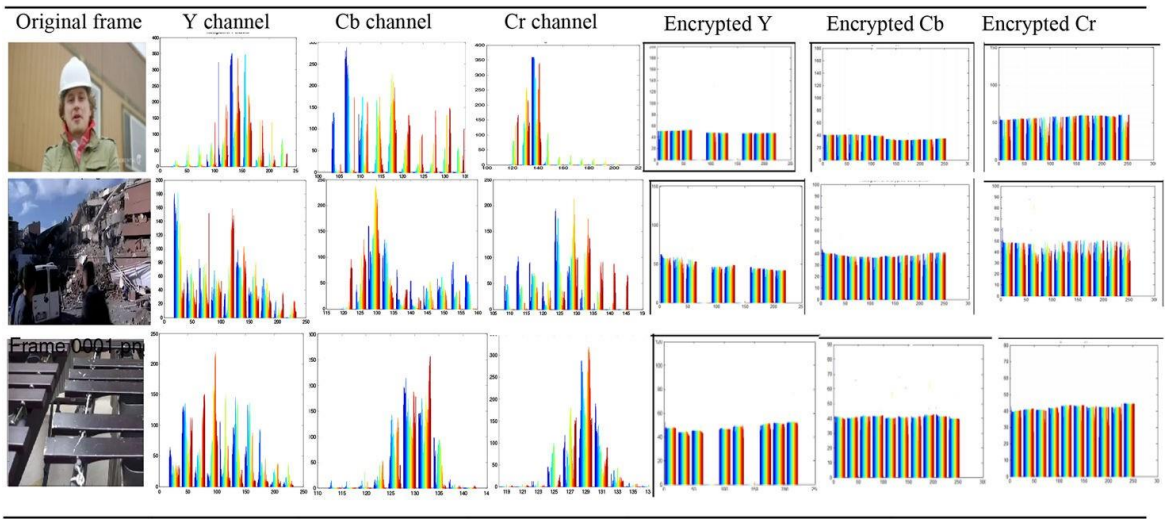

Fig. 2 *Histogram analysis of test videos before and after encryption using VCMTH algorithm*

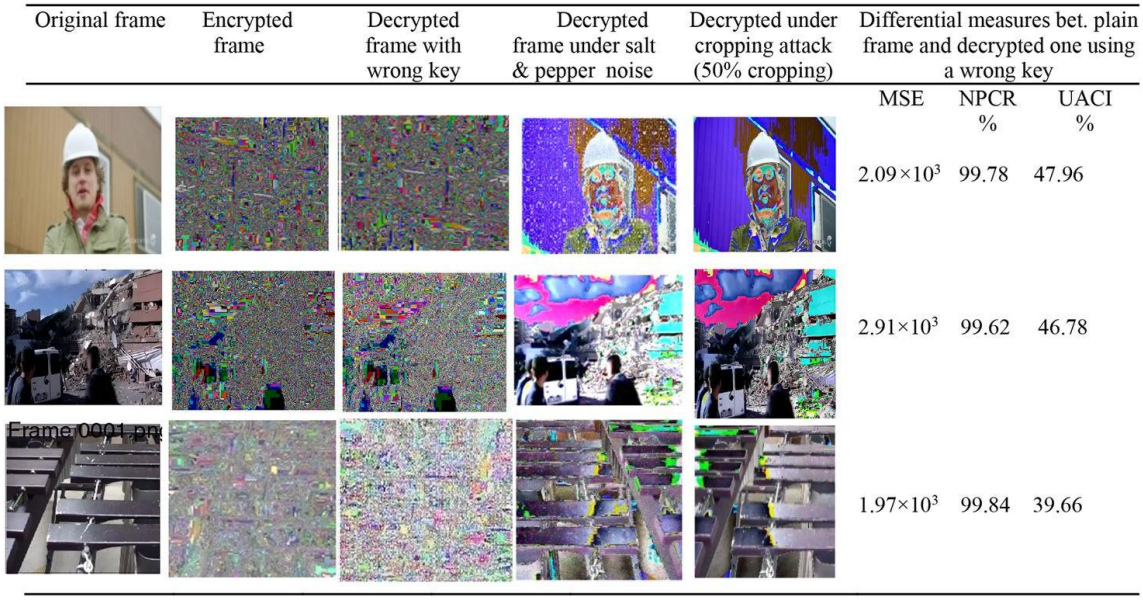

Fig. 3 *Encrypted frames, incorrectly decrypted frames and differential measures for VCMAB algorithm*

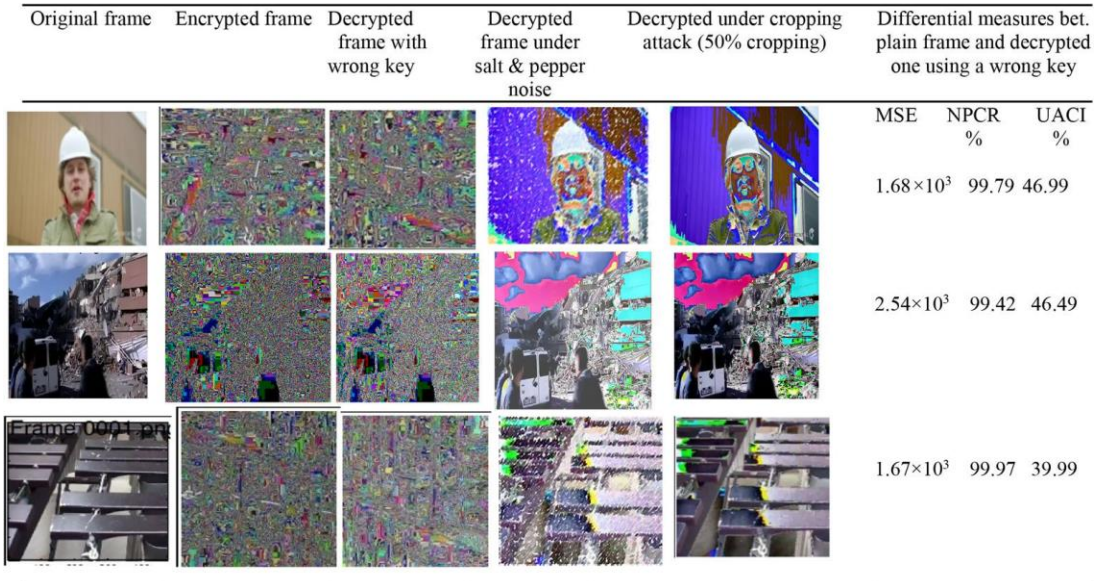

Fig. 4 *Encrypted frames, incorrectly decrypted frames and differential measures for VCMTH algorithm*

tampered with during transmission. The cropping attack can also be detected as apparent from the same two figures.

6.1 Numerical measures

In Table 1, it is apparent that the second algorithm produces decrypted video files with slightly higher quality as dictated by the higher PSNR values, which is a desirable feature. However, the higher PSNR values for the encrypted files are a weakness point regarding security. Additionally, in this table, we show the effect of varying the size of the block involved in the shuffling process.

Smaller block sizes yield higher PSNR values in the reconstruction process and lower PSNR for the encrypted frames.

In Table 2, the mean correlation coefficients between adjacent pixels along the horizontal, vertical and diagonal directions are shown for a video frame before and after encryption. We can easily see that both of the proposed encryption algorithms successfully de-correlate the neighbouring pixels in the decompressed but encrypted video frame. Moreover, it also shows that larger block sizes used in the shuffling process give slightly higher correlation coefficients.

Table 1 Foreman video PSNR for encrypted and decrypted videos for both algorithms based on different block sizes

Block size	PSNR (dB) for decrypted video (VCMAB)	PSNR (dB) for encrypted video (VCMAB)	PSNR (dB) for decrypted video (VCMTH)	PSNR (dB) for encrypted video (VCMTH)
(1 × 1 blocks)	36.1487	11.9547	37.9875	12.6598
(4 × 4 blocks)	34.1578	12.5478	37.1058	13.9789
(8 × 8 blocks)	32.8874	12.9988	35.1120	14.1232
(16 × 16 blocks)	31.0012	13.2146	33.9812	14.9887

Table 2 Average correlation coefficients of original/encrypted videos for both algorithms based on

Original video	Original frames average pixel correlation coefficients			Average pixel correlation coefficient	Encrypted frame		
	Horz.	Vert.	Diag.		Horz.	Vert.	Diag.
frame 1 foreman (VCMAB)	0.9951	0.9760	0.9905	16×16 blocks	0.0019	0.0045	0.0033
				8×8 blocks	0.0017	0.0039	0.0021
				4×4 blocks	0.0009	0.0025	0.0018
				1×1 blocks	0.0005	0.0009	0.0014
frame 1 foreman (VCMTH)				16×16 blocks	0.0025	0.0037	0.0049
				8×8 blocks	0.0018	0.0024	0.0039
				4×4 blocks	0.0014	0.0021	0.0026
				1×1 blocks	0.0008	0.0011	0.0014

Table 3 PSNR decrypted videos under cropping and salt and pepper noise attacks using both algorithms

Video	VCMAB		Algorithm	VCMH		
	PSNR (dB) centercropping	PSNR (dB) sidecropping	PSNR (dB) salt PSNR (dB) center and pepper noise cropping		PSNR (dB) sidescropping	PSNR (dB) salt and pepper noise
Foreman	26.54	23.78	29.45	29.99	25.81	30.01
Aircrash	28.47	24.15	27.61	29.47	25.98	29.47
Xylophone	26.98	25.99	26.99	27.78	26.94	27.99
Balcony	27.79	25.88	28.78	28.97	26.49	29.47

Table 4 Differential measures of some test videos using both algorithms

Encryption algorithm	Original video (file name)	MAE	NPCR%	UACI%
VCMAB	frame I-Foreman	86.6952	98.6327	34.0338
	frame II-Aircrash	85.0917	98.4548	33.4646
	frame IV-Xylophone	85.3743	98.7410	35.3258
VCMTH	frame I-Foreman	87.3515	99.3783	35.1427
	frame II-Aircrash	85.7136	99.0278	34.0279
	frame IV-Xylophone	86.7077	99.1077	35.6591

Table 5 Mean encryption time in seconds for both VEAs based on different block sizes

Encryption algorithm	Original video	Encryption time 1×1 block	Encryption time 4×4 block	Encryption time 8×8 block	Encryption time 16 × 16 block
VCMAB	Foreman	0.9874	0.8145	0.7154	0.6015
	Aircrash	0.8798	0.7471	0.6415	0.5749
	Xylophone	0.6198	0.4012	0.2215	0.1978
VCMTH	Foreman	0.7849	0.6541	0.5987	0.4512
	Aircrash	0.6741	0.5497	0.3418	0.3011
	Xylophone	0.4854	0.3987	0.2457	0.1987

Table 3 shows the negative impact of both the salt and pepper attack and the cropping attack on the PSNR for the decrypted video frames. The VCMAB algorithm is more sensitive to such attacks as indicated by the lower PSNR values compared to those obtained for the VCMTH algorithm.

The mean absolute error between the pixels values of an original frame and the decompressed but encrypted frame is shown in Table 4 for both algorithms. Both algorithms give high MAE values. Other differential measures showing the sensitivity of both algorithms to slight changes in a video frame are also shown in this table.

6.2 Speed analysis

The proposed encryption algorithms have been implemented using MATLAB 2013 on a personal computer with Intel i2-duo 2.20 GHz processor and 3 GB RAM running Windows 7. Table 5 shows the mean encryption time for a video frame using both algorithms for different videos and using different block sizes. Increasing the block size reduces the running time. It is also shown that the running time of the VCMTH algorithm is shorter than that of the VCMAB algorithm.

Table 6 PSNR for both VEAs variants based on Feistel structure

Video	VCMAB		VCMTH	
	PSNR (dB) for decrypted video	PSNR (dB) for encrypted video	PSNR (dB) for decrypted video	PSNR (dB) for encrypted video
Foreman	32.89	13.58	33.78	14.49
Aircrash	31.98	13.99	33.98	14.87
Xylophone	34.47	14.65	36.73	13.89
Balcony	32.95	13.79	34.48	13.96

Table 7 PSNR for decrypted videos under cropping and salt and pepper noise attacks using both algorithms variants based on Feistel structure

Video	VCMAB*			VCMTH*		
	PSNR (dB) centre cropping	PSNR (dB) sides cropping	PSNR (dB) salt and pepper noise	PSNR (dB) centre cropping	PSNR (dB) sides cropping	PSNR (dB) salt and pepper noise
Foreman	28.14	26.19	27.65	29.12	27.98	28.98
Aircrash	26.98	25.74	27.84	27.18	27.99	28.14
Xylophone	27.94	24.87	28.17	29.48	26.83	29.46
Balcony	24.85	26.48	25.99	26.58	28.51	26.56

Table 8 Differential measures of some test videos using both algorithms variants based on Feistel structure

Encryption algorithm	Video frame	MAE	NPCR%	UACI%
VCMAB*	frame I-Foreman	85.9874	97.9871	33.1324
	frame II-Aircrash	84.9487	97.9861	32.9876
	frame IV-Xylophone	85.9987	97.6978	35.9874
VCMTH*	frame I-Foreman	86.9841	99.4123	34.9465
	frame II-Aircrash	84.7894	98.9878	33.7891
	frame IV-Xylophone	84.9867	99.4587	33.6598

Table 9 Mean encryption time in seconds for both VEAs variants based on Feistel structure

Original video	VCMAB*	VCMTH*
Foreman	1.98	1.84
Aircrash	1.78	1.12
Xylophone	0.987	0.874

7. Decission

In this section, two aspects of the proposed algorithms are investigated. First, the impact of using a Feistel structure in the substitution step on both the security and the running time of the algorithms is examined. The second aspect is the applicability of both algorithms with a different compression technique.

In an attempt to improve the security of the proposed algorithms, we tried replacing the simple XORing step by a computationally efficient block cipher scheme named SIT, which has been introduced in [30]. This block cipher involves a key of 64 bits and operates on a plaintext block of 64 bits as well. It consists of five rounds. Each round involves some logical operations, shifting, swapping and substitution. The algorithm is a mixture of a Feistel structure and a uniform substitution-permutation network. This scheme is resistant to linear and differential attacks. When applying this variation to the first algorithm, now denoted as VCMAB*, the 64-bits key is extracted as a pre-agreed subset of the bits of the encrypted pre-selected frame. On the other hand, when this variation is applied to the second scheme, denoted as VCMTH*, the 64-bits key is generated using Henon's map.

The results of applying this variant of the proposed algorithms are shown in Tables 6–9. The low PSNR values for the encrypted frames in Table 6 and the high values for the differential measures in Table 8 again conform with the desired properties of a secure video encryption scheme. Moreover, the distortion due to noise attacks indicates that tampering with the video during transmission is detectable as demonstrated in Table 7. However, the use of a Feistel structure with several rounds makes the algorithms slower as clear from the information given in Table 9. The mean encryption time of a frame is more than doubled. Yet, if a larger block size is used in the shuffling process, it can help reduce the computational time. For instance, the mean encryption time of a frame of the Xylophone video reduces to only 0.6147 s using VCMTH* with a 16×16 block size. For real-time applications, the inexpensive hardware implementation described in [30] could be used to speed up the encryption/decryption process and maintain the desired high level of security.

To demonstrate that the proposed algorithms indeed provide a framework compatible with almost any video compression algorithm, we tried using LZW algorithm as a lossless compression technique [26]. In this case, the quality of the original video is preserved. A sample of the results obtained using the LZW compression algorithm instead of the MPEG-2 compression is shown in Table 10. The obtained differential measures are comparable with the former results presented in Table 4. It is noteworthy that using this lossless compression technique had a negative impact on the execution time.

8. Comparative Study

To further investigate the performance of the proposed schemes, we compare it with other schemes in the literature.

Al-Hayani *et al.* in [31] proposed an integrated scheme for both video compression and encryption. They proposed a hybrid compression algorithm that combines DWT, DCT and vector quantization. The encryption algorithm involves three secret keys fed to two LFSRs

used to shuffle the DWT coefficients several times. Only level 3 DWT coefficients are encrypted. It is noteworthy that the results presented in their paper were obtained for the videos after conversion to greyscale. For a fair comparison, we applied our schemes to the greyscale version of the videos. It is clear from Table 11 that the three schemes produce high quality decrypted videos as dictated by the high mean PSNR values. Moreover, all of them succeed in producing low mean PSNR values, as seen in Table 12, for the encrypted frames, owing to significant distortion caused by encryption. Our schemes have the advantage of being more flexible since traditional compression techniques with efficient hardware and/or software implementations can be used without any changes made to them. This makes their integration into a communication system more straightforward. Additionally, selective encryption still leaks some information about the original video that can be exploited by attackers.

Table 10 Differential measures of some test videos using both algorithms variants based on lossless compression

Encryption algorithm	Video frame	MAE	NPCR%	UACI%
VCMAB	frame I-Foreman	87.1254	98.9841	33.9871
	frame II-Aircrash	86.9691	98.8769	34.9886
	frame IV-Xylophone	86.9988	98.1499	35.9874
VCMTH	frame I-Foreman	88.9141	99.1449	32.9865
	frame II-Aircrash	87.9194	98.9878	33.8891
	frame IV-Xylophone	87.4789	99.1477	34.9468

Table 11 Comparison between the proposed schemes and the scheme in [31] based on mean PSNR for decrypted videos Algorithm

	VIP lane departure		Video Xylophone		Rhinos	
	STD	Mean	STD	Mean	STD	Mean
VCMAB	0.85471	37.6587	0.32548	34.4897	0.6547	36.9854
VCMTH	0.76578	36.9877	0.4987	35.9873	0.59874	34.1549
Al-Hayani *et al.* [31]	0.6552	36.8360	0.2940	31.2958	1.6364	34.4655

Table 12 Comparison between the proposed schemes and the scheme in [31] based on mean PSNR for encrypted videos

Algorithm	VIP lane departure		Video Xylophone		Rhinos	
	STD	Mean	STD	Mean	STD	Mean
VCMAB	0.21547	13.9657	0.04878	13.9578	0.3954	14.3257
VCMTH	0.29875	14.1257	0.05125	13.4599	0.3214	15.9651
Al-Hayani *et al.* [31]	0.3196	13.2366	0.0569	13.2158	2.1767	13.1353

Table 13 Comparison between the proposed schemes and scheme in [32] based on differential measures

Algorithm	Rhinos(Frames 41) (Frames 4)		Video VIP lane departure	
	NPCR	UACI	NPCR	UACI
VCMAB	99.1587	34.5746	98.9897	33.9845
VCMTH	99.8791	35.8465	99.2364	34.6487

Abbas and Shibeeb [32]	99.6615	33.5298	99.6544	33.4134

Table 14 Comparison between the proposed schemes and the scheme in [33] based on PSNR of encrypted frames

Algorithm	Video
	News
VCMAB	13.9847
VCMTH	15.1423
Alhassan *et al.* [33]	15.1122

In [32], a modification to the AES algorithm is made to suit video encryption that employs the Henon map to generate a chaotic mask. The encryption key consists of the 128 bits of the AES algorithm, in addition to 64 bits for the initial seed of the Henon map. It is worth noting that their work does not consider the impact of compression on the results, which is an essential component to provide efficient video transmission. In Table 13, the differential measures for our schemes are compared with those obtained using the scheme in [32]. All schemes yield similar results indicating sensitivity to changes in the key and the plain video frame.

A perceptual video encryption scheme, which selectively rearranges blocks of pixels either column-wise or row-wise in a chaotic manner to degrade the visual quality of the source video, is presented in [33]. The scheme involves four keys representing the rotation angle, the number of iterations, the block size and the unit anti-diagonal matrix. The authors argue that their scheme is resistant to known/chosen plaintext attacks. The Feistel structure- based variant of our algorithms provides similar security guarantees. However, the authors do not provide enough analysis of their scheme under different compression techniques. In Table 14, the PSNR of encrypted frames for a sample video are shown using our proposed algorithms and the algorithm in [33], demonstrating their comparable performance in this respect.

Table 15 provides a comparative study between our proposed schemes and the three schemes in [31–33]. From this comparison, it is clear that the weakest scheme from the security perspective is the scheme in [33], where only permutations are used with a relatively small keyspace size. However, the scheme is computationally efficient so it may find use in applications requiring mild security using devices of limited processing power. The rest of the schemes have large enough keyspaces to combat exhaustive search attacks. It is noteworthy that our schemes enjoy the property of being compression-independent and thus can be easily integrated within existing systems.

The proposed schemes are less computationally efficient compared to the schemes in [31, 33], but have similar efficiency to the scheme in [32]. However, the proposed schemes are flexible and one may tradeoff between security and speed. For instance, only I-frames may be encrypted. Moreover, to improve the computational efficiency of the proposed schemes, the new locations of the frames blocks as dictated by the shuffling map can be pre-computed and stored in look-up tables if enough storage is available. In this case, our schemes would provide both high level of security in addition to being computationally efficient.

Table 15 Comparative study between the proposed schemes and the schemes in [31–33]

Algorithm	Compression	Aspect Keyspace size	Type of encryption
VCMAB	compression-independent	$2^{30\times 10}63$	full
VCMTH	compression-independent	10^{100}	full
Al-Hayani *et al.* [31]	joint compression-encryption	10^{161}	selective
Abbas and Shibeeb [32]	—	2^{234}	full
Alhassan *et al.* [33]	—	2^{33}	perceptual

9. Conclusion

In this paper, two new video encryption schemes have been proposed. A compression step is employed prior to encryption in both the schemes. Though MPEG-4 is expected to give competitive results, however, we preferred using MPEG-2 in our implementation due to its simplicity and ease of programming. Moreover, it still finds use in TV broadcasting. Furthermore, we considered the use of lossless compression to demonstrate that the proposed schemes are general and that their security is independent of the compression technique employed.

According to our simulation results, the first scheme (VCMAB) is slower, however, it is more resistant to CPA compared to the second scheme (VCMTH). Moreover, both schemes produce decrypted videos of high quality as indicated by the high PSNR values obtained. Additionally, the encrypted videos when decompressed show quite low PSNR values emphasising that the proposed schemes conceal the information carried in the videos from attackers. This is further elaborated on through the uniform histograms of the encrypted frames and the low correlation coefficients among adjacent pixels. Furthermore, the proposed schemes are sensitive to slight variations in the decryption key. Any small change in a component of the decryption key yields a totally corrupted video frame. This is both pictorially demonstrated and using numerical measures such as MSE, NPCR and UACI.

The keyspace for both algorithms is shown to be large enough to withstand exhaustive search attacks. The performance of both schemes is analysed under cropping attack and salt and pepper noise attack. The decrypted videos are distorted and thus the receiver will detect that the video data has been maliciously tampered with during transmission by an active attacker. Additionally, resistance to linear and differential cryptanalysis attacks can be achieved by using a Feistel structure in the substitution step of both algorithms which has a cheap hardware implementation to enhance the speed of the algorithm.

The running times of both schemes can be made quite small by choosing a suitable block size for use in the permutation step and consequently are suitable for real-time applications.

A comparative study with recent schemes in literature reveals that our schemes provide superior security levels at the cost of increased computational time. However, the use of pre-computed look-up tables based on chaotic maps can greatly enhance the speed of the proposed schemes.

10. References

[1] Abomhara, M., Zakaria, O., Khalifa, O.: 'An overview of video encryption techniques', *Int. J. Comput. Theory Eng.*, 2010, 2, (1), p. 103

[2] Liu, F., Koenig, H.: 'A survey of video encryption algorithms', *Comput. Secur.*, 2010, 29, pp. 3–15

[3] Effelsberg, W., Steinmetz, R.: '*Video compression techniques*' (dpunkt- Verlag, Heidelberg, 1998)

[4] Salomon, D.: '*Data compression: the complete reference*' (Springer Science & Business Media, New York, 2004)

[5] Zeng, W., Lei, S.: 'Efficient frequency domain selective scrambling of digital video', *IEEE Trans. Multimed.*, 2003, 5, (1), pp. 118–129

[6] Xie, D., Kuo, C.-C.J.: 'Multimedia encryption with joint randomized entropy coding and rotation in partitioned bitstream', *EURASIP J. Inf. Secur.*, 2007, 2007, article ID: 35262

[7] Meyer, J., Gadegast, F.: 'Security mechanisms for multimedia data with the example MPEG-1 video'. Project Description of SECMPEG, Technical University of Berlin, 1995

[8] Qia, L., Nashrstedt, K.: 'Comparison of MPEG encryption algorithms', *Comput. Graphics*, 1998, 22, (4), pp. 437–448

[9] Tang, L.: 'Methods for encrypting and decrypting MPEG video data efficiently'. ACM Int. Conf. on Multimedia, Boston, MA, USA, November 1996, pp. 219–229

[10] Shi, C., Bhargava, B.: 'A fast MPEG video encryption algorithm'. Proc. 6th ACM Int. Conf. on Multimedia, New York, USA, 1998, pp. 81–88

[11] Shi, C., Bhargava, B.: 'An efficient MPEG video encryption algorithm'. Proc. IEEE Symp. on Reliable Distributed Systems, West Lafayette, USA, 1998, pp. 381–386

[12] Shi, C., Wang, S.Y., Bhargava, B.: 'MPEG video encryption in real-time using secret key cryptography'. Int. Conf. on Parallel and Processing Techniques and Applications (PDPTA'99), Las Vegas, NV, USA, June 1999, pp. 2822– 2828

[13] Li, Z., Wang, X., Lin, Y.,: 'RDEA: a novel video encryption algorithm', in (Jong Hyuk) Park, J.J., et al. (Eds.): '*Advanced multimedia and ubiquitous engineering*', (Springer, Berlin, 2015, vol. 352), pp. 183–189

[14] Wu, C.-P., Kuo, C.-C.J.: 'Design of integrated multimedia compression and encryption systems', *IEEE Trans. Multimed.*, 2005, 7, (5), pp. 828–839

[15] Socek, D., Magliveras, S., Culibrk, D., et al.: 'Digital video encryption algorithms based on correlation preserving permutations', *EURASIP J. Inf. Secur.*, 2007, 2007, Article ID 52965, p. 15

[16] Liu, F., Koenig, H.: 'Puzzle – a novel video encryption algorithm'. IFIP Int. Conf. on Communications and Multimedia Security (CMS), Salzburg, Austria, 2005 (LNCS, 3677), pp. 88–97

[17] Malladar, R., Sanjeev Kunte, R.: 'Selective video encryption based on entropy measure', in Krishna, A.N., et al. (Eds.): '*Integrated intelligent computing, communication and security*', (Springer, Berlin, 2019, vol. 771), pp. 603–612

[18] Zhang, X., Seo, S.H., Wang, C.: 'A lightweight encryption method for privacy protection in surveillance videos', *IEEE. Access.*, 2018, 6, pp. 18074–18087

[19] Poynton, C.: '*Digital video and HDTV algorithms and interfaces*' (Morgan Kaufmann,

Publishers, San Francisco, USA, 2003)

[20] Jack, K.: '*Video demystified: a handbook for the digital engineer*' (Elsevier, Oxford, 2005)