Journal of Engineering Science and Technology
Vol. 15, No. 5 (2020) 3375 - 3387

A NEW TAMPER DETECTION ALGORITHM FOR VIDEO

DUJAN B. TAHA[1], SHAHD A. HASSO[1], TAHA B. TAHA[2,*],

[1]Software Engineering Department,
College of Computer Science and Mathematics, Mosul University, Iraq
[2]IT Department, Faculty of Sciences, Tishk International University, Erbil, Iraq
*Corresponding Author: PhD.Taha@gmail.com

Abstract

Few years ago, authentication of visual media has regarded as a crucial research area which concerns with the development of methods and tools to determine if the digital media has been forged or not. The detection of video tampering was a concern of many researchers because of its important role in road accident, court events, and other applications. In this manuscript, a new low cost video forgery detection algorithm has been proposed by utilizing the correlation coefficients between the video frames and embedding them as an encrypted data into the first frame of the video stream. Experimental results show the high performance of the proposed algorithm regarding visual quality and robustness due to the ability to detect tampering even in simple and low-effect attacks.

Keywords: AES algorithm, Video authentication, Video tampering.

1. Introduction

Nowadays, video tampering becomes a serious problem due to the easy synthesis of fake photographic images to promote a story through media channels and social media. This is due to great development in computer graphics and animation applications, and the availability of many digital image processing and manipulation tools with low cost. With the lake of suitable regulatory frameworks and the appropriate infrastructure to prosecute such sophisticated cybercrime, there is a growing discontent with the increasing use of these low enforcement tools. Therefore, there is a need to authentic video files [1-3].

Any video application can contain three parts: A creator, a recipient, and a third party. The creator produces the video, and the recipient receives the video from the creator through the third party. The third could be a storage device (e.g., CD/DVD) or it could be a noisy channel in a video stream. In addition, if the recipient sends the video to another party after receiving it, the recipient can also be a third party. Malicious attackers point to such third parties to change the content of video. Video authentication means determining that a particular video content is real and exactly the same as it was at the time of capture. Video authentication actually means content authentication since the recipient must not get a precise copy of the original video without any modification. For example, because of the large storing capacity, video has to be compressed, and the common video compression techniques (e.g., MPEG 1, 2, 4) is loss. Of course, the decompressed video is different from the original video. Yet, it should be still considered authentic.

A video file generally consists of a container that contains video data represented by a video coding format and audio data represented by an audio coding format. The container may as well encompass synchronization data, headings and metadata (such as titles). Standardize video file types, such as (.webm), are configuration files that are specified by limiting which container format and which are the allowed formats for audio and video compression. The coded video and audio in the container of a video file are named essential elements. A program that decompresses video or audio is usually termed as a codec.

A good design generally indicates that the file extension allows users to export from the file extension the program that can open the file. Such video file formats are Windows Media Video (.wmv), WebM (.webm), Ogg Video (.ogv) and Flash Video (.flv). Each of these formats may contain only a few videos and audio coding format which makes it easy to identify what codec will open the file. On the other hand, some common container forms (such as QuickTime (.mov) and AVI (.avi)) encompass almost any video and any audio format, and have a file extension name next to the type of container, which it makes using of the file extension by the end user very difficult to derive the codec or the program used to play the file [4].Video tampering may be as old as the video technique itself. Today, powerful software for scene editing has made it extremely easy to initiate a believable change of the video frame even for non-professionals. Many of tampered media contain a part of the medium that is changed using objects in the same media [5]. All different types of video tampering are usually classified to intra-frame or inter-frame tampering.

(a) Intra-frame tampering: Actual contents of certain frames are altered. Examples of these forgeries are the followings:

- Copy-paste forgeries: In this type, an attacker may include or eliminate an object to or from a video frame.
- Upscale-crop: these forgeries require cropping a video frames to get rid of evidence of the video tampering, and then expanding the modified frames to keep the same resolution through the entire video.

(b) Inter-frame tampering: affect in some way the sequence of frames in a video. Typically, these forgeries involve removing a set of frames or entering them from or in a video. Duplicate a frame is again a type of this counterfeit which can be mentioned as inter-frame copy-paste tampering. Figure 1 presents different types of video forgeries in this category [2].

Digital videos, unlike images, have no fixed specifications which mean probably there are different locations and sizes for video objects even for the same stream with the same format. For this reason, fewer methods have been developed than those developed for image application [1]. In This paper a new low cost video forgery detection algorithm have been proposed by utilizing the correlation coefficients between the adjacent video frames and embedding them as encrypted data into the first frame of the video stream. The paper is presented as follows; next section presents the Advanced Encryption Standard Algorithm. Section 3 demonstrates a literature review of recent video tampering detection methods. Section 4 described the new proposed method, Section 5 exhibits the experimental results. Finally, the paper is concluded in Section 6.

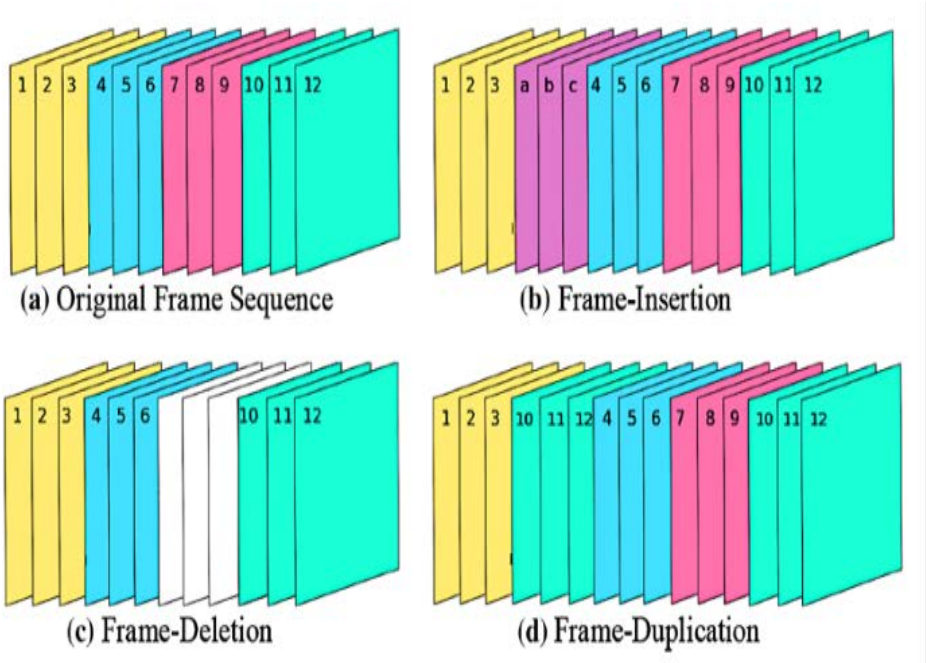

Fig. 1. Inter-frame video forgeries.

2. Advanced Encryption Standard Algorithm

The Advanced Encryption Standard (AES) was considered in 2001 by the National Institute of Standards and Technology (NIST). It is a symmetric block encryption

algorithm replaces DES for many applications. Unlike the public-key encryption algorithms like RSA, the AES structure are very complex and cannot be easily explained like many other cipher algorithms [6, 7].

Encryption receipts a block size plaintext of 16 bytes or 128 bits. Length of the used key can be 16, 24, or 32 bytes (128, 192, or 256 bits). Depending on the key length, AES is named as AES-128, AES-192 or AES-256.

The encryption process consists of N rounds, depending on the key length: 10 rounds for a key of 16 bytes, 12 rounds for a key of 24 bytes, and consists of 14 rounds for a 32 bytes key. First N-1 rounds involve four different conversion functions; Sub Bytes, Shift Rows, Mix Columns and AddRoundKey. Three conversions only are contained in the last round with one initial shift (AddRoundKey) prior to the first round. As inputs, each transform accepts one or more 4×4 matrices. A 4×4 matrix is produced as output. The function of key expansion creates $N + 1$ circular keys, each of them is a matrix of 4×4. AddRoundKey transform accepts each round key as one of its inputs in each round. General steps for the algorithm are:

Step 1: Key Expansion: From the cipher key, round keys are derived considering Rijndael's key schedule.

Step 2: Addition of first round key:

- AddRoundKey: by means of x or, each byte of the round key is joint with a byte of the state.

Step 3: 9, 11 or 13 rounds:

- Sub Bytes: a substitution process. Using a lookup table, each byte is substituted with another byte.
- Shift Rows: a transposition process. Cyclically shifting the latest three rows of the state a certain number of steps.
- Mix Columns: Combining the four bytes in each column of the state.
- AddRoundKey

Step 4: Final round (10, 12 or 14 rounds):

- Sub Bytes
- Shift Rows
- AddRoundKey

3. Literature Review

Existing forgery detection methods can be classified into the common way listed in Table 1. Digital image tampering detection utilizes active or passive detection techniques. Active detection approaches aim to insert a digital watermark in the images when they are taken [3]. Arab et al. [1] used watermarking to detect tampering [1], with respect to the video file format Audio/Video Interleaved (AVI). Two new spatial domain watermarking schemes have been proposed. The representation of each pixel was achieved by 2 bytes in the format of AVI video. To determine which block is changed, a block-wise technique has been used.

Chetty et al. [3] proposed an algorithm based on the transformation of features in cross-modal space and the multimodal fusion of variant kinds of features. These features are obtained from different intra/inter-frame pixel sub-blocks in video streams to detect any video alteration. Proposed algorithm models allowed detecting the tamper in videos

with low bandwidth. This achieved by using passive tamper detection methods and attempting to model signatures that have been embedded in the pre-processing series in the camera. Two different features were evaluated, noise and quantization for copy and move tampering. These features demonstrate the performance of the proposed passive tamper detection.

Table 1. Video tampering detection methods.

Classification	Techniques
Active techniques	Use a tracking model as a watermark which is inserted at the time of recording or when sending the video.
Passive techniques	Determine the contents legitimacy without relying on outside information.
Intra frame techniques	Considering one frame at a time at the analysis process
Inter frame techniques	Neighbouring frames relations are used to detect tampering.
Detection techniques	Detect the existence of alterations without their exact location.
Localization techniques	Locate tampering in addition to detection.

Upadhyay and Singh [4] introduced some issues in the design of the system of video authentication. These issues comprise the classification of tamper attacks, tampering attack types and robustness. Wei et al. [5] proposed an algorithm which gets the visual content scales of the video frame by Gaussian pyramid transform and find the visual content single-scale similarity. Normalized mutual information between two frames has been defined using information theory. To detect the video tampering location, the local outlier isolated factor detection algorithm has been used. Yin et al. [8] used A SIFT (Scale-invariant feature transform)-based function to perform tamper detection in the camera. The SIFT algorithm has been improved to quickly generate a SIFT key point, then the SIFT-based image descriptor had been developed to represent images. Xiaoling and Huimin [9] have developed an algorithm that considers key values as the watermarking. These values were calculated using Hash transform to P-frame table. The watermark was embedded to motion vectors of P-frame. Wang et al. [10] proposed inter-frame tamper detection model when the Consistency of Correlation Coefficients of Gray Values (CCCoGV) has been used as a medical property. In original videos, CCCoGV stayed steady, intended alteration in the frame stream results unrealistic values.

Xia et al. [11] detected duplicate frames. Similarities between and inside frames in the spatial correlations have been identified. Using average texture variation (ATV), a two-step blind detection algorithm for video has been proposed. Value of ATV is calculated for each frame to get the ATV curve of the video. Then, the obtained curve is further treated to highlight its property, which estimates the original frame rate. Qian et al. [12] proposed an algorithm consisted of feature extraction and unusual point localization. In feature extraction process, the 2D phase congruency of each frame has been extracted. Then, using k-means clustering method, the unusual points were detected.

The novelty in this work is to find with a low cost the relationship between adjacent video frames, encrypt the obtained values, and hide the encrypted values in the video file itself. In

the event of any attack on the video, the attacker cannot know the locations of the concealment and if it happened, he cannot decipher the code using AES algorithm because he does not know the key used in the encryption process. In previous studies, attacks have been mentioned with the possibility of their discovery. Perceptual quality and robustness values were not mentioned. They did not give a detection percentage if the change was within one frame that could be included in the video. Various attacks that could be occurred to the video were taken into consideration in the proposed algorithm. Such attacks were revealed with MSE and PSNR calculations for each of them.

4. Proposed Method

A new algorithm has been proposed in this section to detect any modifications to the video. Figure 2 presents the main steps of the proposed algorithm.

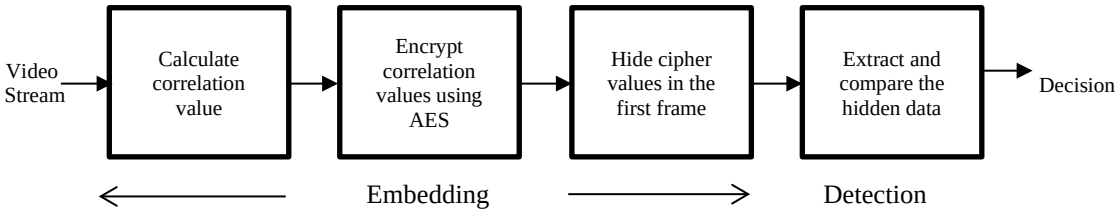

Fig. 2. Basic steps of the proposed algorithm.

4.1. The embedding process

The embedding process consists of three main steps (Fig. 3); calculating the correlation values, encrypting the resultant coefficient values using AES algorithm, and hiding the outcome of the encryption using bit substitution technique. The correlation coefficients are calculated between each two adjacent frames, starting from the second one. Consider a frame (*f*) of a video sequence of *n* frames. The frame correlation coefficient (frame-corr) is defined as:

$$frame-corr(f_i, f_{i+1}) = \frac{\sum\sum(f_i, f_{i+1})}{\sqrt{(\sum\sum(f_i + \bar{f}_i)^2)(\sum\sum(f_{i+1} + \overline{f_{i+1}})^2)}} \quad (1)$$

$$for \quad i = 2 \quad to \quad n-1, \quad \bar{f}_i \quad and \quad \overline{f_{i+1}} \quad are \quad means \quad of \quad f_i \quad and \quad f_{i+1}$$

Correlation values for adjacent frames are encrypted using AES-128 algorithm. Encryption receipts a block size of 128 bits or 16 bytes plaintext with 128 bits key length which is randomly generated. Encrypted correlation values are then embedded into the two least significant bits of randomly selected locations at the first frame.

4.2. The detection process

At the recipient side (Fig. 4), the receiver calculates the correlation values of the sent video, extracts the encrypted correlation values from the random location in the first frame, and decrypts these values using the symmetric key of the AES algorithm. Then, the receiver compares the obtained values. If they match, then there are no tampering, otherwise tampering is detected.

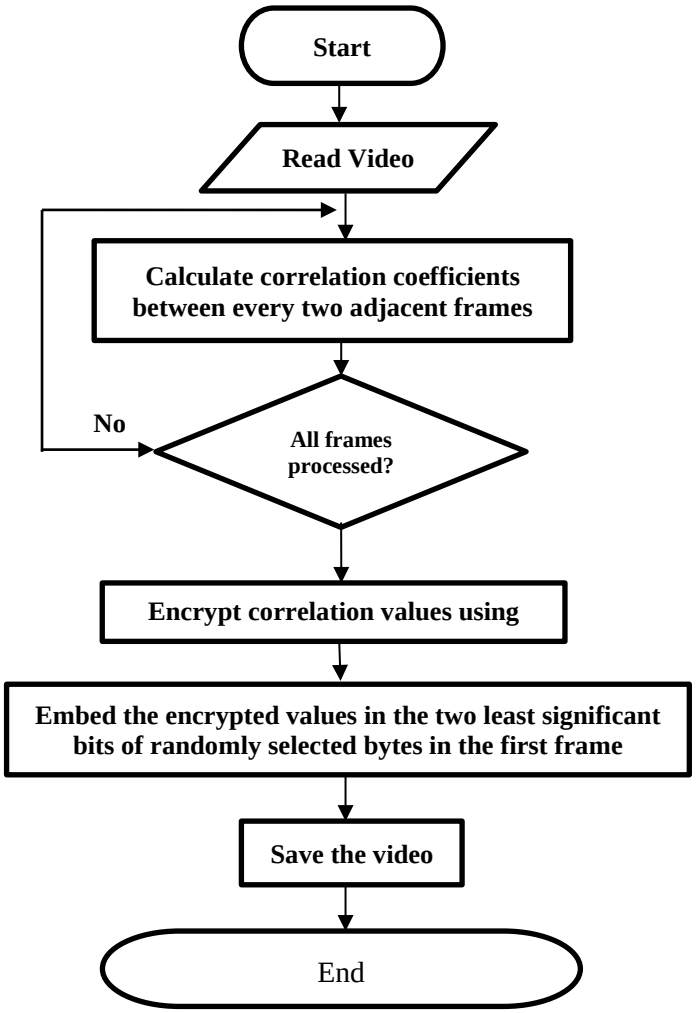

Fig. 3. The embedding process.

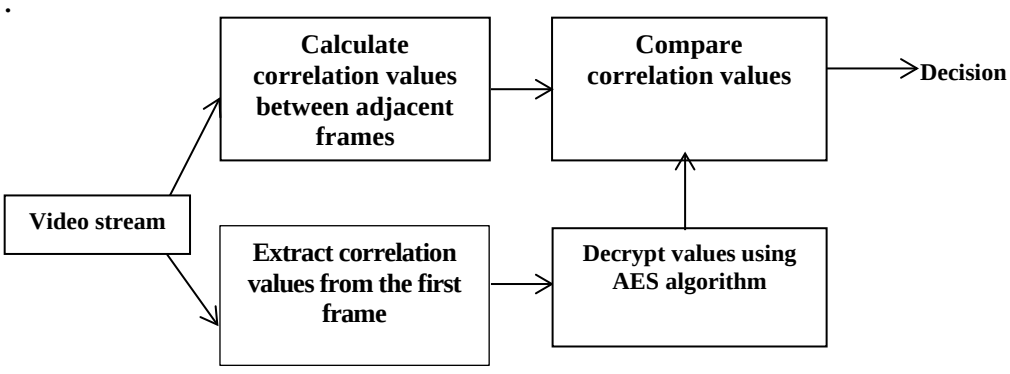

Fig. 4. The detection process.

5. Results

The proposed algorithm can be applied to videos with any extension. In this work, AVI, and .MP4 file types have been used. In order to test the performance of the proposed algorithm, it has been applied to five videos with a different number of frames and different frame sizes. Perceptual quality and robustness tests were the two important issues considered in results.

5.1. Perceptual quality test

Mean Square Error (MSE), Peak-Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [13] have been used for measuring frame quality after the embedding process. These metrics are defined as follows:

$$MSE = \frac{1}{3MN}\sum\sum\sum_{1}^{3}\left(output\quad video\quad frame - input\quad video\quad frame\right)^{2} \tag{2}$$

$$PSNR = 10\log 10\left[\frac{\max\quad value^{2}}{\frac{1}{3MN}\sum\sum\sum_{1}^{3}(output\quad video\quad frame - input\quad video\quad frame)^{2}}\right] \tag{3}$$

MSE and PSNR are calculated using Eqs. (2), and (3) respectively where the value 3 in summation represents color values of the image. In 24-bit images, each pixel is represented by three color values, namely; red, green, and blue. Therefore, each frame in the video is a 24-bit image. To calculate MSE and PSNR, size of the whole frame need to be obtained according to the following equation:

$$\text{FrameSize} = \text{row}\times\text{column}\times 3 \tag{4}$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5}$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for the input video frame x, and output video frame *y*. SSIM values greater than 0.95 indicate the high perceptual quality in the objective evaluation [13].

Table 2 and Figs. 5 to 7 show the high quality of the resultant frame after the embedding process. Results also demonstrate that when frame size increased, MSE decreases, PSNR increases, and SSIM becomes in its maximum values. Consequently, metrics values depend on the frame size rather than the number of video frames. Better values of these metrics result from large frame sizes and small number of frames. With the increase of the number of video frames and the size of each frame, the execution time increases due to the process of calculating the value of correlation coefficients between each two adjacent frames.

Table 2. Perceptual quality measurements values after the embedding process.

Video no.	No. of frames	Frame size	MSE	PSNR	SSIM
Video1	133	1080*1920*3	0.0014	76.7481	1.0000
Video2	141	240*320*3	0.0422	61.8758	0.9992
Video3	387	202*360*3	0.1182	57.4034	0.9982
Video4	614	1080*1920*3	0.0066	69.9378	0.9999
Video5	747	480*640*3	0.0413	61.9713	0.9992

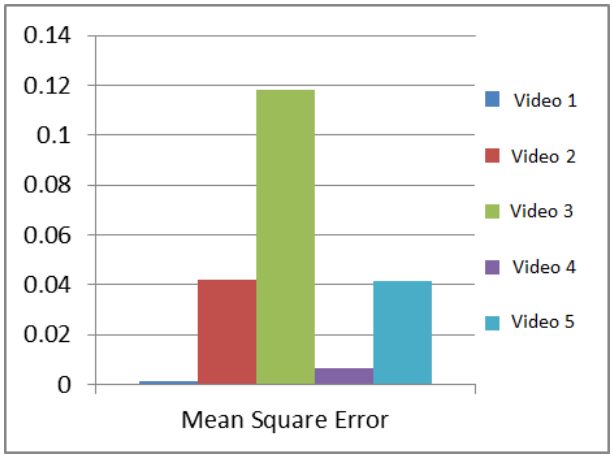

Fig. 5. Mean square error values.

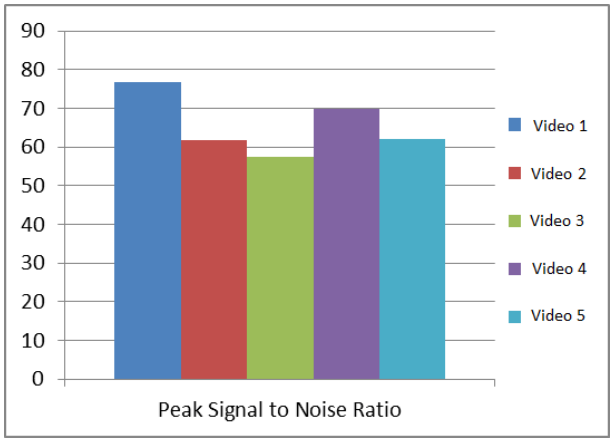

Fig. 6. Peak signal to noise ratio.

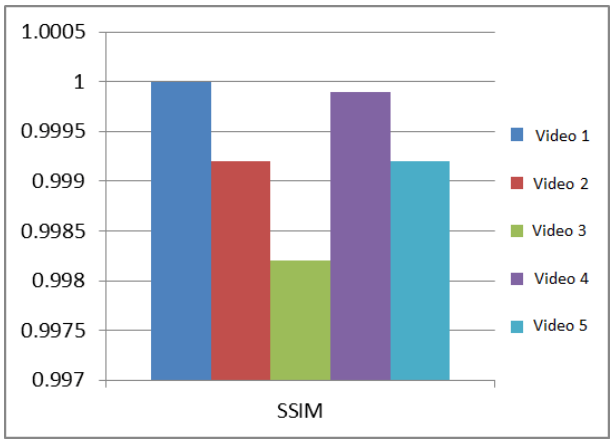

Fig. 7. SSIM values.

5.2. Robustness test

Table 3 shows the result of applying different inter-frame and intra-frame attacks to video 2. Figures 8 and 9 demonstrate the comparison of metric values between the original and attacked video. Values of MSE and PSNR have changed after applying the attacks to the said video file. For example, when updating the frame in Fig. 10 by adding some dots, values of MSE are slightly increased. Consequently, PSNR decreased. In this figure we notice that in spite of adding only two dots to the frame and it looks close to the original frame, values of MSE and PSNR has been changed, which means even small changes can be detected. When duplicating two specific frames, error rate became more than its value before this attack, but less than its value when adding the dots because the added frames are from the same video stream. As for deleting one particular frame, MSE increased due to changing the correlation coefficient values.

When modifying a part of the frame (changing the colour of one column of the piano in Fig. 11), MSE has also become high compared to its value in other attacks. The highest MSE were obtained when exchanging two frames (Fig. 12) because changes in four correlation value of frames have been occurred. Note that an attack with a small change in a frame leads to a change in the value of MSE (even it is small) and this is enough to detect tampering.

Table 3. Robustness values after applying attacks on video 2.

Attacks	MSE	PSNR
Modifying frame (Intra-frame attack)	0.0649	60.0057
Duplicating two frames (Inter-frame attack)	0.0672	59.8551
Deleting a frame (Inter-frame attack)	0.0665	59.9030
Exchanging frames (Intra-frame attack)	0.0671	59.8613
Modifying a part of a frame (Intra-frame attack)	0.0655	59.9661

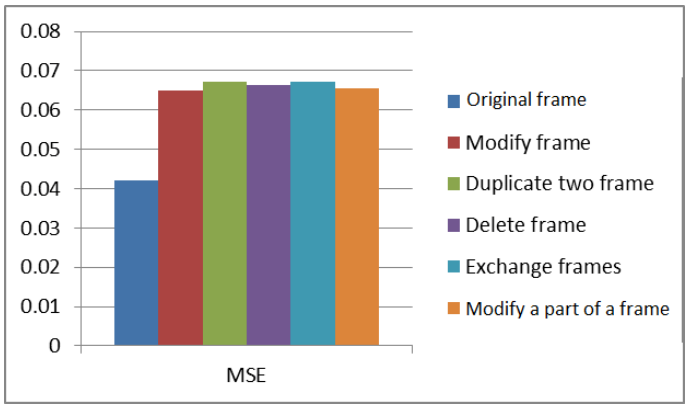

Fig. 8. Comparison of MSE values on original and attacked videos.

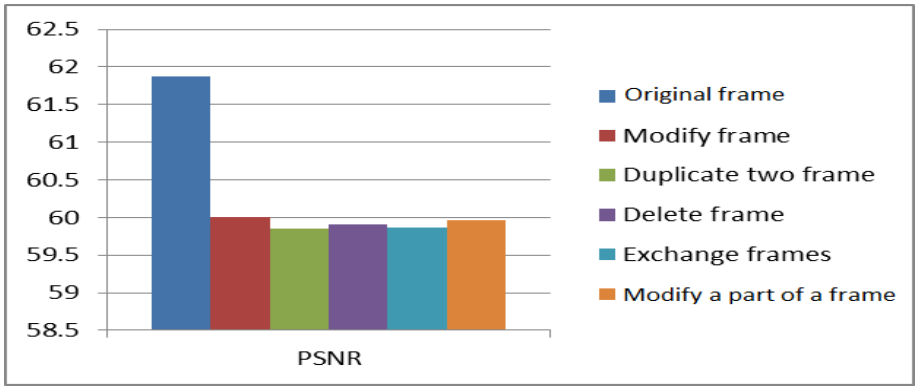

Fig. 9. Comparison of PSNR values on original and attacked videos.

(a) Original frame. **(b) Modified frame.**

Fig. 10. Modifying a frame by inserting some dots.

(a) Original frame. **(b) Modified frame.**

Fig. 11. Modifying a part of a frame.

(a) Original frame.

(b) Modified frame.

Fig. 12. Exchanging frames.

Conclusion

In this paper, a new video tampering detection algorithm has been proposed by finding the correlation coefficients between each two adjacent frames and embedding them into a randomly selected locations of the first frame. For security enhancement, the correlation coefficients were encrypted using AES algorithm before the embedding process. Experimental results show high detection capabilities even after applying different inter-frame and intra-frame attacks while keeping high visual frame quality. On-going research is to detect the exact tamper location, within the frame or within the whole video stream.

References

1. Arab, F.; Abdullah, S.M.; Hashim, S.M.; Manaf, A.A.; and Zamani, M. (2016). A robust video watermarking technique for the tamper detection of surveillance systems. *Multimedia Tools and Applications*, 75(18), 10855-10885.
2. Singh, R.D.; and Aggarwal, N. (2018). Video content authentication techniques: a comprehensive survey. *Multimedia Systems*, 24(2), 211-240.
3. Chetty, G.; Biswas, M.; and Singh, R. (2010). Digital video tamper detection based on multimodal fusion of residue features. *In 2010 Fourth International Conference on Network and System Security* (606-613).
4. Upadhyay, S.; and Singh, S.K. (2012). Video authentication: Issues and challenges. *International Journal of Computer Science Issues*, 9(1), 3.
5. Wei, W.; Fan, X.; Song, H.; and Wang, H. (2019). Video tamper detection based on multi-scale mutual information. *Multimedia Tools and Applications*, 78(19), 27109-27126.
6. Stallings, W. (2017). *Cryptography and network security principles and practice* (7th ed.). England: Pearson Education.
7. Padate, R., & Patel, A. (2014). Encryption and decryption of text using AES algorithm. *International Journal of Emerging Technology and Advanced Engineering*, 4(5), 54-9.

8. Yin, H.; Jiao, X.; Luo, X.; and Yi, C. (2013). Sift-based camera tamper detection for video surveillance. *In 2013 25th Chinese Control and Decision Conference (CCDC)*. 665-668.
9. Xiaoling, C.; and Huimin, Z. (2012). A novel video tamper detection algorithm based on semi-fragile watermarking, *Advances in Information Technology and Industry Applications*, LNEE 136, 489-497.
10. Wang, Q.; Li, Z.; Zhang, Z.; and Ma, Q. (2014). Video inter-frame forgery identification based on consistency of correlation coefficients of gray values. *Journal of Computer and Communications*, 2(04), 51.
11. Xia, M.; Yang, G.; Li, L.; Li, R.; and Sun, X. (2016). Detecting video frame rate up-conversion based on frame-level analysis of average texture variation. *Multimedia. Tools Applications*, 72(1), 1-23.
12. Qian, Li.; Rangding W.; and Dawen X.(2018). An inter-frame forgery detection Algorithm for surveillance video. *Information*, 9(11).
13. Taha, T.B.; Ngadiran, R., and Ehkan, P.(2018). Adaptive image watermarking algorithm based on efficient perceptual mapping model. *IEEE Access*, 6, 66254-66267.