An Image Encryption Scheme for JPEG and M-JPEG Based on Permutation of Pixel-Blocks

Farzad Zowghi, Alireza Keshavarz-Haddad
School of Electrical and Computer Engineering
Shiraz University, Shiraz, Iran
farzadzoghi@gmail.com, keshavarz@shirazu.ac.ir

***Abstract*—In this paper, we propose a new scheme for encryption of commonly used JPEG images and Motion-JPEG (M-JPEG) videos. Most of the related works for video encryption revolve around the idea of efficient and secure encryption of JPEG images and on other video formats such as MPEG2 and MPEG4. Our new method, unlike the existing methods, treats M-JPEG as a video, i.e., a sequence of consecutive JPEG frames, and uses its properties to moderate the effect of encryption on the size of compressed data. For encryption, the presented scheme uses some rotational permutations on blocks of pixels. Unlike most existing techniques, our scheme is not sensitive to sparse transmission errors and also keeps the computational complexity very low.**

Keywords— M-JPEG Encryption; JPEG Encryption; Pixel Permuation; Block Rotation;

I. INTRODUCTION

Multimedia data is often of large volume and needs to be compressed before storage or transmission. Various methods are used to compress images, among which, Joint Photographic Experts Group (JPEG), is one of the widely used coding standard. JPEG is a compression method based on transformation, which adopts the sensitivity of the human eye to reduce the redundant image data. The transformation used in this compression is Discrete Cosine Transform (DCT) [1]. The compression ratio is defined as:

$$CR = \frac{S_{UNCOMPRESSED}}{S_{COMPRESSED}} \quad (1)$$

where S denotes the size of the file. From the definition, CR factor increases as the data gets more compressed.

Digital videos are constructed of consecutive frames of images; therefore, video compression is based on image and data compression techniques. In practice, to handle the large amount of video data – most of which are redundant – different kinds of compression techniques have been developed. The simplest technique is to encode each image one by one, which is called Motion JPEG (M-JPEG). The main disadvantage of M-JPEG is that the relations between frames are not taken into account. The more proper way is to differentiate the constant parts from moving parts of frames and combine the DCT transformation and motion coding to have a more compressed video. The standards include H.263 [2], MPEG-2 [3], MPEG-4 [4], and MPEG-4 AVC/H.264 [5]. Among them, H.264 is vastly used in surveillance systems and IP cameras since it provides high compression ratio. The main disadvantage of this standard is its high complexity and large delay for encoding and decoding processes. For real-time and low complexity applications M-JPEG is more efficient; although it uses more bandwidth due to lower compression ratio. Besides, M-JPEG frames are encoded individually and if one frame does not get thorough due to transmission errors, it will not affect other frames. This means that M-JPEG is a proper streaming technique when data is transported over noisy communication links.

In this paper, we first propose an encryption scheme for JPEG files based on permutation on blocks of pixels. The scheme slightly changes the size of JPEG images, but it has very low complexity compared to the existing joint compression and encryption methods. The scheme can be implemented on a minimal hardware with memory and shift-registers for performing some permutations and circular shifts on the pixels of an image. Next, we develop an encryption scheme for streaming M-JPEG videos based on the proposed scheme. The new one uses another compression technique to reduce the size of M-JPEG video further. The main advantage of this scheme is that the effect of transmission error on the decrypted video in the receiver side will be minimal. Moreover, the scheme has a very low computational complexity. In addition, the proposed scheme has flexibility to properly set its parameters based on the desired compression and noise level in the transmission links.

In the course of this paper, we will discuss previous and related methods used for JPEG and M-JPEG encryption in section II. After that, in section III, some metrics are defined to assess the security and efficiency of encryption algorithms. In section IV, our new method is proposed and evaluated. Then, we explain some simulation results in section V, and finally, conclude the paper in section VI.

II. RELATED WORKS

There are various encryption techniques for multimedia systems. Encryption of multimedia content including audio, image, and video can be classified as below:

A. Complete Encryption

This method encrypts the content as bits/bytes of data using standard encryption algorithms such as DES, AES, and RC4.

This operation can be performed on raw and compressed data as well. Images and videos usually contain great amount of redundancy; hence, they can be expressed in smaller sizes. Compression algorithms are designed based on the statistical properties of raw multimedia content. On the other hand, encryption algorithms try to mix the data in a way that they seem random to anyone who is not authorized to have access to them causing higher entropy and larger size. Thus, if the encryption is performed before the compression the result will have a very low CR.

When the encryption is performed after compression, the format of the image or video is encrypted as well. As a result, if a part of the data gets noisy or even get lost during transmission, recovering the whole data will become very difficult or even impossible in some cases. To resolve this issue, some sequence numbering and proper frame formatting must be used for sending the data. These parameters create new overhead on the system which is not suitable in some practical applications.

B. Selective Encryption

Selective encryption algorithms encrypt only a part of the image or video and leave the rest unchanged. Selection of these parts is typically based on their impact on visual perception of the image/video. Most selective encryption schemes for JPEG images encrypt a selected number of DC or AC coefficients when the image is transformed into the frequency domain using DCT [6], [7], [8]. Some other methods encrypt a selected part of raw image data such as, human skin or face [9]. Based on these ideas, authors in [10] proposed a scalable encryption for M-JPEG video streams in which various portions of frames are encrypted.

Note that some of these techniques require high computational complexity, since they need to jointly compress and encrypt the multimedia messages. These methods are mainly used for commercial applications that the security of the encryption scheme is not of top priority. In other applications such as military, unintelligibility of all parts of the image is of high importance. For instance, people or objects and their movements and even the background must be completely meaningless to human eye. Selective encryption techniques could not achieve this level of security without a big rise in their encryption complexity.

C. Permutation Algorithms

These algorithms could be performed on raw image pixels or compressed bytes of data. Permutation for compressed data has to interchange bytes or other type of blocks of data with each other. In this technique, the video or image becomes a low quality version of the original one, as only a few bytes are interchanged [11] and the video or image is still clear to human eye. In this paper, we perform permutations on raw image pixels.

1) Random pixel permutation: Random pixel permutation changes the position of pixels under the control of a pseudo-random number generator. The implementation of this algorithm has a low computational cost. A great part of our proposed method lies under this class and as a consequence, security and efficiency of these algorithms will be discussed later.

2) Chaotic map-based algorithms: A chaotic map is a system denoted by mathematical equations with an initial value as input and control parameters that determine the action and the output to be a random sequence. The output sequence is very sensitive to the initial value and control parameters. For example, a secure efficient algorithm based on 3D baker map has been proposed in [12] and an encryption scheme based on multiple chaotic maps is presented in [13].

Note that as we explained earlier, if an independent compression algorithm is used after the encryption, the size of the output file can become much larger. In this case, we introduce a new encryption scheme in which the size of compressed file gets slightly larger.

III. SECURITY AND EFFICIENCY METRICS

The objective of encrypting a video or an image is to secure its contents. However, this operation changes some properties of the data such as redundancy that affects the compression. In addition, the encryption process must be of low complexity to be applicable in real-time streaming. Here, we introduce some metrics to evaluate various encryption methods.

A. Cryptographic Security

Cryptographic security is determined by the ability to resist the cryptanalysis methods. The level of security of an encryption algorithm is hard to evaluate, since new cryptanalysis techniques could be developed in the future to break the encryption scheme. Generally, if an encryption algorithm is secure against most of the well-known attacks, we say that the algorithm possesses high security [14].

B. Perceptual Security

An encryption method should make the image or video unintelligible to human eye. Some metrics can be used to measure the quality of an image but not exactly its intelligibility. The typical metric for images is Peak Signal-to-Noise Ratio (PSNR). Although PSNR does not show the perception of the image for a human eye, it can show that the image is unintelligible if be low enough (less than 10 dB) as it is seen in the experimental results in [15].

$$PSNR = 10\log_{10}\frac{L^2}{MSE} \tag{2}$$

where L is the maximum pixel value and MSE (mean square error) satisfies:

$$MSE = \frac{1}{P}\sum_{i=1}^{n}(c_i - p_i)^2 \tag{3}$$

where c_i and p_i are the encrypted and the original pixel values and P is the total number of pixels.

C. Compression Efficiency

Images and videos are being compressed so as to reduce the storage space or transmission bandwidth. Encrypting an image

alters the statistical properties of the data and consequently, changes the effectiveness of compression algorithm. To evaluate the effect of encryption on compression ratio we can define a Changed Compression Ratio (CCR) based on the size of the original compressed image (S_0) and size of encrypted image (S_1):

$$CCR = \frac{S_1}{S_0} \times 100 \tag{4}$$

D. Computational Complexity

In order to encrypt real-time multimedia streams, it is imperative to utilize algorithms with low computational complexity. Encryption always increases the processing time and in some cases requires data buffering in both transmitter and receiver. These procedures should not be time-consuming in real-time applications.

IV. THE PROPOSED ALGORITHMS

In this section, we first propose a method for encryption of JPEG images. Afterwards, we utilize the presented scheme in accompany with a technique in video compression to encrypt M-JPEG video streams efficiently.

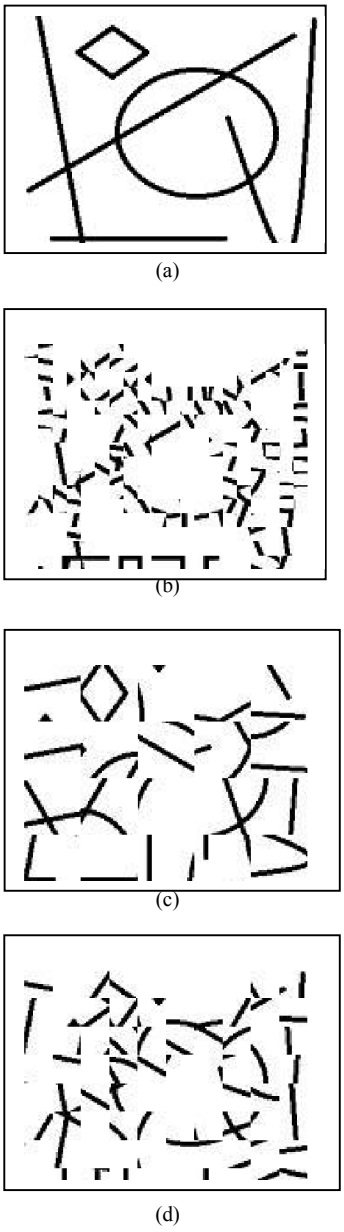
A. Encryption of JPEG images

In our scheme, we divide images into MN squares of size 8×8-pixel blocks. So the frame is considered as a matrix of size $M \times N$. The reason for choosing this size of blocks is that the JPEG standard algorithm performs on 8×8 blocks for compression. The same block size is used to have consistency with JPEG and achieve proper compression efficiency.

Our algorithm permutes blocks of pixels in the image. For the human eye, similar shapes in the background or edges in a picture is more understandable. Therefore, if we just permute the blocks, they will already contain similar shapes and after a while, some of these blocks can be brought back together, like solving a puzzle, and this matter could not be tolerated in high security applications. To prevent this kind of simple but yet feasible attacks, we introduce a *rotation function* that divides the image into $B \times B$-pixel blocks and then *rotates* and *flips* each of these blocks randomly. The quantity of B affects the computational complexity that will be discussed later. Besides, B must be chosen regard to the complexity and bigness of different curves in the picture. It is obvious that choosing small amount for B can still preserve the figure of big objects and selecting a large quantity for B could not change the small shapes and objects. Therefore, B must be chosen adaptively based on the elements in the picture. Fig. 1 depicts the results of rotation function with different values for B.

B. Encryption of M-JPEG videos

There are many encryption algorithms for M-JPEG videos which are based on methods which are used for encrypting JPEG images. These methods treat each frame as a distinct image. Moreover, in our proposed algorithm, we take inter-frame relationships into account. Knowing that frequent use of M-JPEG videos are in surveillance cameras which their frames share a common feature, e.g., an almost fixed background. We try to get use of this information to reduce the CCR but not in a way that causes excessive complexity for real-time applications.



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

Fig. 1. (a) The original image (b) Effect of using small quantity of B (shape of the circle is understandable) (c) Effect of using large quantity of B (the rhombus can be seen almost entirely) (d) Effect of using proper quantity of B

Here we describe our encryption scheme in seven steps.

- Step 1: For the current frame, if *frame_number* $\equiv$ *1 mod K,* then set this frame as reference frame and go to step 4.
- Step 2: Find the difference between this frame and the latest reference frame.

- Step 3: If the number of nonzero elements in the difference is larger than the threshold μ, then set this frame as a new reference frame.
- Step 4: Use the *rotation function* on the frame with parameter B for the block size.
- Step 5: Randomly choose n rows which were not chosen previously and permute them circularly with shifts randomly picked between 1 and $M-1$.
- Step 6: Randomly choose m columns which were not chosen previously and permute them circularly with shifts randomly picked between 1 and $N-1$.
- Step 7: If all rows and columns have been chosen, go to the next frame; otherwise, go to step 5.

Note that m and n are defined based on the aspect ratio of the video, e.g. for QCIF, CIF, and 4CIF formats m is 11 and n is 9 (see Table I). In the presented algorithm, the random choices are based on the output of a pseudorandom generator with a secret key as its seed.

TABLE I. RESOLUTION OF VIDEO FORMATS

Format	Video Resolution	M	N
QCIF	176 × 144	22	18
CIF	352 × 288	44	36
4CIF	704 × 576	88	72

V. ANALYSIS AND SIMULATION RESULTS

In this section, we analyze the performance of the proposed schemes for JPEG images and M-JPEG videos and provide some simulation results.

A. Cryptographic Security

Encryption algorithms that use permutation are often vulnerable to known-plaintext attacks. In these attacks the attacker has a piece of plaintext and compares it to the ciphertext so as to find the key. In order to overcome this problem, we need to change the key frequently or use a sophisticated random generator with very long loop in its output. Any of these can be used to ensure the cryptographic security of the proposed algorithm e.g. changing the input of random generator in some multiples of K.

B. Perceptual Security

There are two types of output frames; independent frames and differential frames. Independent frames which are initial frames of every K-frame set, are encrypted individually and do not need another reference frame for decryption. Other frames are not encrypted directly. Their difference with the reference frame is computed and then this difference in encrypted. Differential frames mostly include zero values because of little difference among frames. Thus, we could correctly assume that they are unintelligible to human eye, as shown in Fig. 3. Although independent frames still have small blocks of the original image, they cannot be intelligible to human eye if been chosen small enough. Here we use 8 pixels by 8 pixels blocks which are smaller than the minimum resolution required for human eye to detect a familiar face [16]. PSNR values for different video formats are computed in Table II.

Furthermore, the random sequence for choosing rows and columns to be permuted and the random shifts are used to ensure that the defined blocks move randomly. Adjacent blocks in the original image get detached in the encryption process with a great probability and the reason for that can be explained by examining the number of moves for blocks. In the process of the algorithm, there are N rows and M columns with M and N blocks in each respectively; which makes the total number of moves $2MN$ and the average number of moves equal to 2. It is clear that blocks move in rows and then columns and so on. Consequently, two nearby blocks, one with even number of moves and the other one with odd number of moves could not get to each other in the procedure. Distribution of number of moves for blocks of a frame is shown in Fig. 2. As we can see, the most probable number of moves is 2. Number of blocks that have 1 or 3 moves is near the amount of blocks with 2 moves. As a result, we can say there is a good chance that neighbor blocks do not lie near each other after the encryption.

In Fig. 3 we compare a selective encryption method with our proposed method for three successive frames. The selective method encrypts all DC coefficients of the image and most part of AC coefficients as well. We can see that in encrypted frames of selective method, shape of the head and the face and some other parts of the background and movements in consecutive frames are still visible. On the other hand, encrypted frames of the proposed method, both independent and differential frames are unintelligible to human eye.

C. Compression Efficiency

Consecutive frames in a video in most cases are very similar to each other. Introduction of differential frames in the algorithm is a direct use of this property. In addition, the reference frame changes once the difference between the current frame and the previous reference frame reaches a level that the subtracted image is not redundant enough to have less size than the original one, e.g. when a sudden movement happens in one frame. It can be seen in Table II that for QCIF, CIF and 4CIF videos we can achieve CCRs lower than 100 percent by choosing right amount for K.

A parameter that affects the compression efficiency of the algorithm is K. The reason is that K determines the number of independent frames in an encrypted video. It is clear that independent frames must have CCRs higher than 100 because of less redundant data which is a result of the encryption process. As a consequence, we expect the average CCR boosts as K gets smaller which is shown in Fig. 4. $K=1$ means every frame is encrypted independently.

Another parameter that affects the CCR is μ, which effect depends on K as well. We should expect μ to have more impact on compression efficiency as K increases. When K is large enough that a big change in video could happen, then the threshold μ can counter the effect of this change on difference of the frames and compressed data size. Smaller μ results in

more differential reference frames and it means that frames will be more similar to reference frames; hence, less compressed size due to more zero pixels. Results are depicted in Fig. 5.

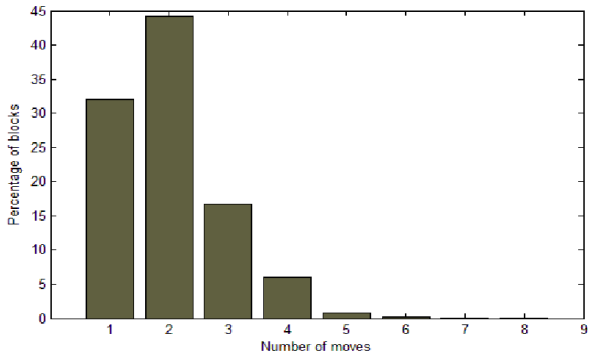
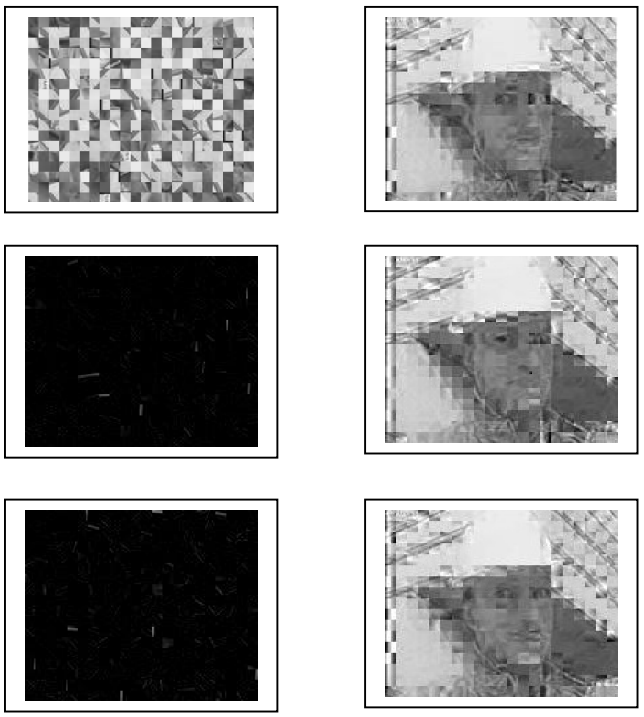


Fig. 2. Distribution of number of moves for different blocks



(a) (b)

Fig. 3. (a) Three successive frames encrypted by the proposed algorithm (the first frame is independent and the two other are differential frames) (b) The same three frames encrypted by the selective encryption method (all DC and most part of AC coefficients are encrypted)

D. Computational Complexity

Computational cost of the proposed algorithm is much lower than MPEG-4 and H.264 standards that take many parameters into account to maximize the compression ratio. We defined parameter *B* in the algorithm which represents the size of the blocks for rotation and flip operations. Although this parameter has very low impact on compression efficiency, it has a direct effect on computational complexity of the algorithm. It is expected to need less processing time as *B* gets larger. This effect is illustrated in Fig. 6 for a CIF video. It can be seen in Table II that for QCIF and CIF videos with 30 frames per second and 16 and 32 as amounts of *B*, the encryption process does not cause a delay and for 4CIF videos there is a maximum of 12 milliseconds of delay which can be considered as real-time streaming with *B=48*. Minimum requirements for our simulations are 1.6GHz CPU and 500MB of RAM.

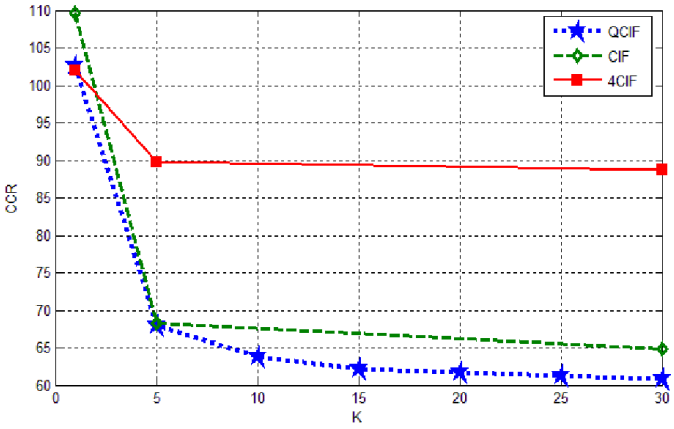


Fig. 4. Effect of changing *K* on CCR with μ=40%

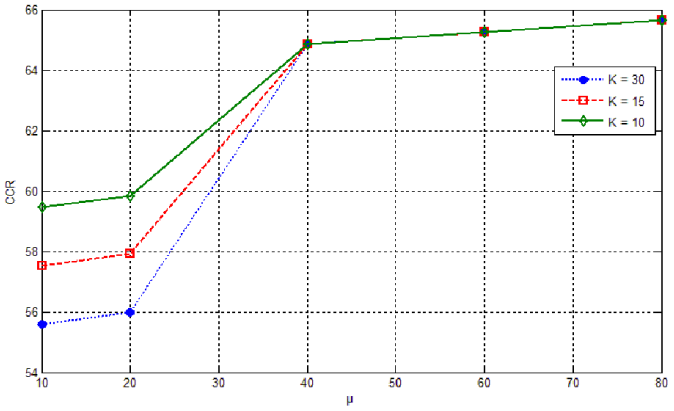


Fig. 5. Effect of changing μ on CCR for vaious *K*

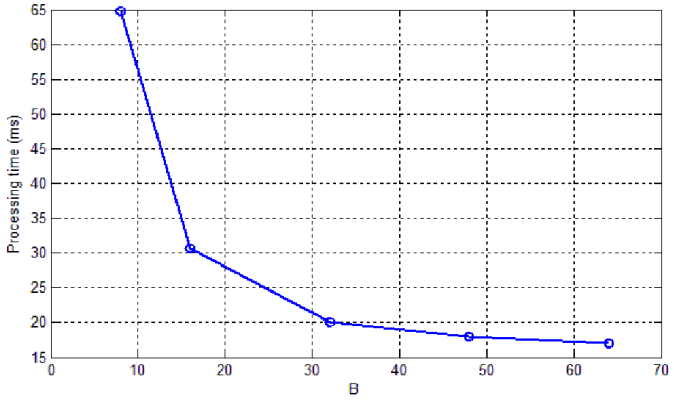


Fig. 6. Effect of changing *B* on processing time of the algorithm for CIF format

E. Decryption

Decryption process is similar to encryption one. For identifying the reference frames in decryption, we use the same threshold and among every consecutive K frames, there is an independent frame as the first reference. In this procedure, the permutations of rows and columns and then the rotations and flips of $B \times B$ blocks must be in reverse order of the encryption process. In order to get the correct sequence of rows and columns, the receiver needs to have the secret key and the algorithm for the random generator used for encryption. For differential frames we need to add the reference frame to the decrypted image to get the original exact frame. Note that employing an incorrect key as seed for random generator results in an incorrect sequence of rows and columns and consequently the decrypted frame will still look like an encrypted frame.

F. Controling the parameters

The parameters defined in the presented algorithm such as B, K, and μ can be changed in order to get to the desired computational complexity and compression efficiency needed for the system. As said before, B must be chosen based on the area and the objects of where the camera is located. In our experiments we used 16, 32 and 48 as amounts of B for QCIF, CIF, and 4CIF videos, respectively. Amount of K must be based on the noise of the channel and the needed compression efficiency. If the transmission links are noisy there is a possibility that a frame does not get to the receiver correctly and the worst case is when this frame is an independent reference frame. In this case, we will lose all the ongoing frames as soon as the next independent frame arrives. The number of lost frames is determined by K. Consequently, it is important to choose K based on this and then change μ to match the needed compression efficiency.

TABLE II. SECURITY AND EFFICIENCY OF THE PROPOSED METHOD

Resolution	Efficiency and Visual Security		
	***CCR*[a]**	***Encryption Process Time (ms)*[b]**	***PSNR (dB)***
QCIF	77%	7-9	4.5
CIF	96%	17-20	4
4CIF	112%	38-45	8

[a] CCR computed with K=30 and μ=*40%*

[b] Processing time computed with B=16 for QCIF, 32 for CIF and 48 for 4CIF

VI. CONLUSION

In this paper, we reviewed various techniques for encrypting video streams and then proposed a new method for encrypting M-JPEG video streams. We used inter-frame relations to alleviate the effect of encryption on data compression. Experiments show that the proposed method is proper in terms of computational complexity for surveillance videos. Common sizes for M-JPEG videos used in IP cameras are 160×120 or 320×240 which have lower resolution than QCIF and CIF samples used in our simulations. Moreover, in our method independent reference-frames help the receiver to becomes synchronized with the sender and assure that there will be less than K frames loss if one of the reference frames gets lost during transmission. This illustrates that our proposed scheme is a proper choice for real-time M-JPEG streaming over noisy channels.

REFRENCES

[1] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", 3rd ed, Prentice-Hall, 2008.

[2] "ITU-T Recommendation H.263-Video Coding for Low Bit Rate Communication". version 1, November 1995; version 2. January 1998

[3] ISO/MPEG-2. "ISO 13818-2: Coding of moving pictures and associated audio", 1994.

[4] MPEG-4 part 2. "ISO/IEC 14496-2: Advanced Simple Profile (ASP)".

[5] H.264/MPEG-4 Part 10. "ISO/IEC 14496-10: Advanced Video Coding (AVC)", ITU-T H.264 standard.

[6] Osama A. Khashan, Abdullah M. Zin, Elankovan A. Sundararajan, "Performance study of selective encryption in comparison to full encryption for still visual images," Journal of Zhejiang University-SCIENCE C (Computers & Electronics), pp. 435-444, 2014.

[7] M. Van Droogenbroeck and R. Benedett, "Techniques for a selective encryption of uncompressed and compressed images," in Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium, Sept. 2002.

[8] T. Stutz and A. Uhl, "Format-compliant encryption of H.264/AVC and SVC," Proceedings of IEEE International Symposium on Multimedia, pp.446-451, Dec. 2008.

[9] Jose Rodrigues, William Puech and Adrian Bors, "Selective encryption of human skin in JPEG images," IEEE International Conference on Image Processing (ICIP'06), pp. 1981-1984, Oct 2006.

[10] Lei Chan, N. Shashidhar and Q. Liu, "Scalable secure M-JPEG video streaming," 26th International Conference on Advanced Information Networking and Applications Workshops, pp. 111-115, 2012.

[11] J. Rajpurohit, S. Sharma, B. Naruka, "A comparative study of video encryption schemes," International Journal of Computer Applications, vol. 91, pp. 10-16, April 2014.

[12] Shiyu Ji, Xiaojun Tong and Miao Zhang, "Image encryption schemes for JPEG and GIF formats based on 3D baker with compound chaotic sequence generator," in *arxiv.org,* 2012.

[13] G.A. Sathishkumar, K. Bhoopathy bagan and N. Sriraam, "Image encryption based on diffusion and multiple chaotic maps," International Journal of Network Security and Its Apllications (IJNSA), vol. 3, No. 2, pp. 181-194, March 2011.

[14] Shiaguo Lian, "Multimedia Content Encryption: Techniques and Applications", CRC press, 1st ed, 2009, pp. 8-15.

[15] Lingling Tong, Feng Dai, Yongdong Zhang and Jintao Li, "Visual security evaluation for video encryption," MM '10, Proceedings of the International Conference on Multimedia, pp. 835-838, 2010.

[16] Pawan Sinha, "Identifying perceptually significant features for recognizing faces," Proc. SPIE 4662, Human Vision and Electronic Imaging VII, June 2002.