International Journal of Informatics and Communication Technology (IJ-ICT)
Vol.7, No.1, April 2018, pp. 39~48
ISSN: 2252-8776, DOI: 10.11591/ijict.v7i1.pp39-48

Efficient Encryption Technique for H.264/AVC Videos Based on CABAC and Logistic Map

Fatma K Tabash*, M. Izharuddin
Department of Computer Engineering, Aligarh Muslim University, Aligarh-202002, U.P, India

Article Info

Article history:

Received Jan 17, 2018
Revised Mar 7, 2018
Accepted Mar 13, 2018

Keyword:

CABAC
Encryption technique
Real-time application and robust

ABSTRACT

Nowadays, the demands of real-time video communication are increased rapidly. Search and rescue (SAR) applications like earthquake rescue, avalanche victims, wildfire monitoring in addition to highway surveillance are considered examples of real-time applications. In which, communication time is considered the most important metric to be optimized to ensure support for victims lives. Thus finding a simple and time efficient encryption technique for securing the transmitted data become mandatory. In this paper, we present an efficient encryption technique which has low computation complexity, low processing time and highly chaotic encrypted videos. The proposed technique is based on CABAC where the bin-string of Intra-Prediction Mode is encrypted with chaotic signals and the sign of MVD is toggled randomly. For residue coefficients the sign of the AC coefficients are flipped randomly and the first value of DC coefficients is encrypted by XORing the bin-string with random stream. All random streams are generated with chaotic systems using Logistic map. The experimental results shows that the proposed technique is highly effective for real-time application and robust against different types of attacks.

Corresponding Author:

Fatma K Tabash,
Department of Computer Engineering, Aligarh Muslim University,
Aligarh-202002, U.P, India.
Email: fatma.tabash@gmail.com

1. INTRODUCTION

With the rapid development of internet technology, multimedia data such as images, audios or videos are used more widely in human's daily life. Whereas coping and modifying these data can be easily achieved, Digital rights management (DRM) become an important research field to protect the copyrighted multimedia data. H.264/AVC video is one of the most common formats used for video compression and encoding. It is popularity comes from the saving of 50% of the bitrate required to transmit video stream. Accordingly, it spans a wide range of application starting from low bitrate internet streaming applications to high bitrate HDTV broadcast applications. Safeguarding video streams from unauthorized users become an urgent demand. Encryption techniques are the most commonly used ways to conceal the information of video stream from the obtrusive third party. Encryption techniques for H.264 has many approaches either to encrypt the total bit stream for gaining the maximum security or partially encrypted. Partial/selective encryption approach is commonly used because they do not increase the total bit rate significantly or change the format compatibility. Above all, with partial encryption, total computation cost is also reduced. The full-stream encryption approach is followed when the high security communication is required as in military needs or news scoops, but the drawbacks of these approaches is that they change the format compatibly and increase the bitrate. Most of content provider does not bother to present their media with low quality service, therefore partial encryption approach can effectively be used to protect multimedia data with low bandwidth taxation. In literature, many techniques has been proposed to partially encrypt H.264/AVC video data: Shahid et al. [1] proposed a selective

encryption technique to protect H.264/AVC video based on two entropy coding modules (CAVLC and CABAC) simultaneously. In CAVLC, The signs of trailing ones and the signs of non-zero coefficients are encrypted using AES cipher, where as in CABAC, non-zero coefficients of same length bin-stream in addition to the signs of non-zero coefficients are encrypted with the same cipher. The drawback of this algorithm is that perceptual security is not good i.e. it does not conceal the information of the plaintext. Su et al. [2] proposed another encryption scheme based on IPM, MVD and residue data. IPM data are encrypted by doing XOR between IPM codeword and 3 bits of random stream. For encrypting MVD, only flipping the sign of MVD is considered to keep format compliant and less complexity. In case of residue, the array of coefficients is divided into units of nonzero levels and zero-run pair i.e. nonzero level associated with its preceding zeros. After dividing, a scrambling technique is applied into these units. The drawback of these algorithm is that it increase the bitrate. Yeung et al. [3] proposed different encryption scheme based on new unitary transforms as an alternative to the common integer DCT transform. The disadvantage of this technique is that it needs new designed H.264/AVC encoder which is cost much. Lian et al [8] proposed another partial-encryption technique based on CAVLC. The suffix part of Exp-Golomb code of IPM is encrypted with a random stream cipher. The signs of MVD and AC coefficients also encrypted. DC coefficients are encrypted differently: in CAVLC, the signs trailing ones T1 and levels are encrypted. But encrypting the sign of levels in CAVLC is increasing the bitrate because they are used in encoding process.

2. BACKGROUND

2.1. Chaotic Systems

In the recent years, use of the chaotic theory is growing rapidly to implement encryption process. Although the traditional ciphers like DES, IDEA, RSA and AES are much secured and most commonly used in encryption algorithms, they are very complex and needs high computational cost. For this, they are more suitable for binary/ text application where the size of the plaintext is considered moderate. For multimedia applications especially for videos the size of data for processing is very huge, encrypting them with traditional cipher will produce very significant overhead in processing time. In addition, the high-redundancy between two consecutive frames makes traditional ciphers fails to hide the significant information of the source file. For these reasons, there has been a significant trend towards other ciphering systems that has low computational complexity and simple implementation. Chaotic systems has been proposed to be an efficient encryption technique for real-time multimedia applications. There are many reasons to use chaotic system as an alternative to traditional ciphers: time evolution of the chaotic signal strongly depends on the initial conditions and the control parameters of the generating functions: slight variations in these quantities yield quite different time evolutions. Consequently those initial conditions and the control parameters can be used as a shared encryption key.

Different types of chaotic maps are used for encryption process. Logistic Map is one the simplest and very secured functions. Formula (1) shows mathematical form of Logistic Map:

$$x_{n+1} = r x_n (1 - x_n) \quad (1)$$

Where it depends on the initial condition x_n and one control parameter r. The initial condition x_n can be any floating point number between 0 and 1 and the control parameter has range from 3.7 to 4.

$$0 < x_n < 1 \text{ And } 3.7 \leq x_n < 4$$

In our work we select Logistic Map for encryption process because it is very simple and much suitable for real time applications.

2.2. Entropy Coding Process of IPM, MVD and Residue Coefficients in CABAC

2.2.1. Entropy coding for IPM (Intra-Prediction Modes)

To elaboration of the encoding process for IPM will be done over three parts: first, explanation of Syntax Elements (SE) that carry the values of IPM is presented. Second, different binarization processes in CABAC is taken . Finally, dedication of SEs with their suitable binarization process is achieved.

Part 1:

In CABAC, IPM for 4x4, 8x8 and chroma blocks are encoded in the bit-stream using five syntax elements:

prev_intra4x4_pred_mode_flag: this flag is enabled, if IPM value of the current 4x4 block equals the minimum mode of the upper and left neighboring blocks.

rem_intra4x4_pred_mode: this syntax element holds the value of the best IPM mode in the current 4x4 block.
prev_intra8x8_pred_mode_flag: this flag is enabled, if IPM value of the current 8x8 block is equal the minimum mode of the upper and left neighboring blocks.
rem_intra8x8_pred_mode: holds the value of the best IPM mode in the current 8x8 block.
intra_chroma_pred_mode: holds the value of the best chroma-prediction mode.

In case of 16x16 block texture, the entire 16x16 macroblock will be considered as one block for prediction. Therefore, only one value of prediction mode considered for the entire macroblock. For this reason, there is no designated syntax element to hold the value of Intra_16x16 prediction mode. The value of Intra_16x16 prediction mode will be included implicitly under the syntax element **mb_type** (that holds the type of macroblock) alongside with the value of Coded Block Pattern (CBP).

Part 2:
IPM are binarized in CABAC using three types of binarization processes:
Fixed-length (FL) binarization process [7]: FL binarization is constructed by using a fixedLength binary representation of unsigned integer of the syntax element value. fixedLength = $Ceil(log_2^{(cMax+1)})$ where, cMax is the maximum value of unsigned integer represented using binary string.
Truncated unary (TU)[7]: is truncated unary binarization. All values of SE which are less than cMax are binarized using unary binarization. All greater values are binarized same as cMax value.
Unary (U) binarization process[7]: The bin-string is consists of 1s of length equal to the value of SE terminated with zero.

Part 3:
Each SE is binarized with a specified binarization process:
prev_intra4x4_pred_mode_flag: binarized using FL with cMax=1 i.e. one bit binary string to represent SE. If flag is enables The bin-string="1" else it is "0". Same manner is to binarize **prev_intra8x8_pred_mode_flag.**
rem_intra4x4_pred_mode: binarized using FL with cMax=7 i.e. it needs 3 bits to represent the value of SE. Same manner for **rem_intra8x8_pred_mode.**
intra_chroma_pred_mode: binarized using TU of cMax=3 i.e. all value smaller than 3 is encoded using Unary binarization, else bin-string will consists of two ones "11".

2.2.2. Entropy Process for Residue Coefficients

Residue coefficients are encoded in CABAC using the next syntax elements:
significant_coeff_flag: indicates if the current coefficient is significant/nonzero or not. This means, it draw the map for the positions of nonzero coefficients.
last_significant_coeff_flag: indicates the position of the last nonzero coefficient.
coeff_abs_level_minus1: curry the absolute value of the nonzero coefficients minus one.
coeff_sign_flag: curry the sign of nonzero coefficients.

It is noticed that, the encoding process of residual data depends on four different types of SEs which are highly correlated. Accordingly, any encryption process for residual data should be done carefully because any change in these information may lead to break format compatibility of the transmitted source.

Three of SEs: **significant_coeff_flag, last_significant_coeff_flag and coeff_sign_flag** are binarized using Fixed-length (FL) binarization process with cMax=1 because they are flags. So, 1 bit bin-string will be sufficient to hold the value. For **coeff_abs_level_minus1,** *Concatenated unary/k-th order Exp-Golomb(UEGk)* binarization process with k=0 [4] is used.

2.3. Encoding Process for Motion Vector Difference (MVD)

Motion Vector Difference is encoded using two syntax elements:
mvd_l0[ ][ ][0], mvd_l1[ ][ ][0] to curry the values of MVD in X-axis.
mvd_l0[ ][ ][1], mvd_l1[ ][ ][1] to curry the values of MVD in Y-axis.

Encryption process for MVD have to be done carefully because any error in encrypting process will leads to format incompatibility and cannot decode the source file. The binarization process for MVD is done using *Concatenated unary/k-th order Exp-Golomb (UEGk)* binarization process with k=3 [4].

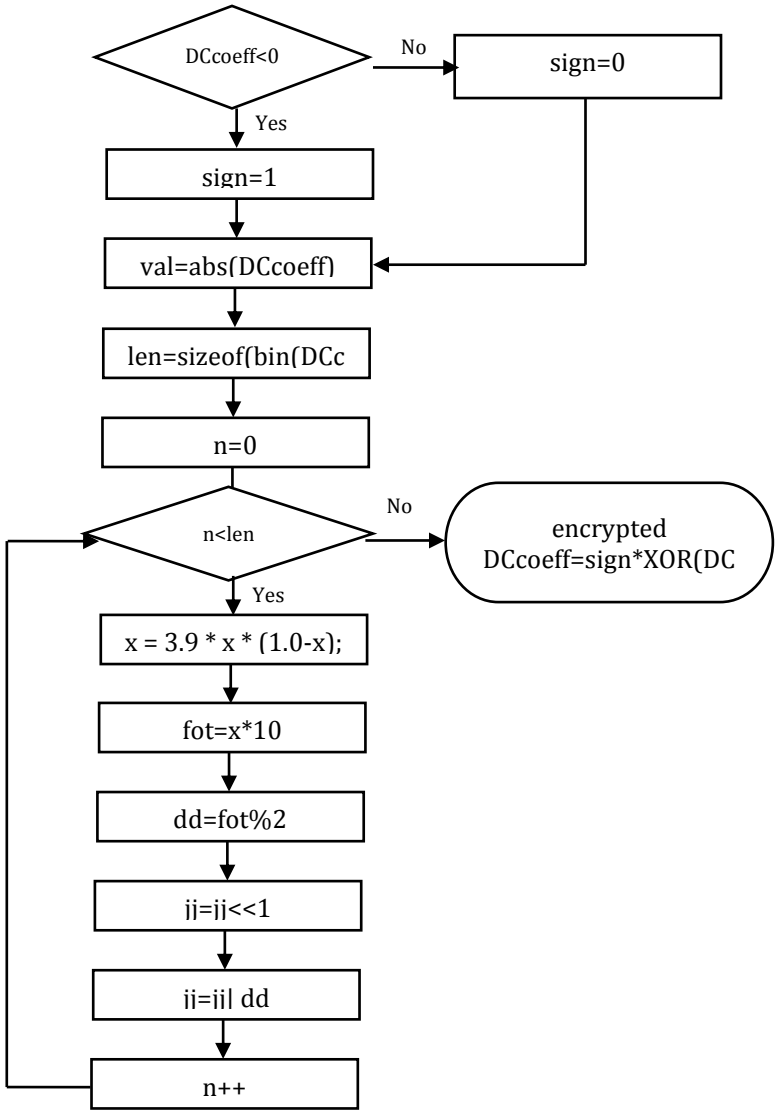

Figure.1 Encryption Process of DC Coefficient

3. PROPOSED SCHEME

The proposed scheme is based on the partial encryption of the three encoding parameters: (i) IPM, (ii) MVD and (iii) residue coefficients. Intra Prediction Mode (IPM) is a good choice for encryption process because the current block is predicted from the edges of the neighboring blocks therefore, encrypting one intra block will propagate the chaos to the other neighboring blocks. The IDR(Instantaneous Decoding Refresh) frame consists of intra-predicted blocks and any P frame predicted using motion compensation will be encrypted using the chaos propagated from the IDR frame. Encrypting only IPM parameter is unsecure because it is vulnerable to replacement attack[5]. For this reason, encrypting residue data will enhance the security of encryption process. Also MVD data will be encrypted to hide the motion information in the video.

Most of the former work is based on traditional ciphers AES, RSA and DES. Despite of their high security measures, they have high computation complexity, which is unsuitable for multimedia application because of large volume of data. For this reason, a different light-weight cipher technique is proposed to encrypt the selected encoding parameters.

3.1. Initialization Process

At the start of encryption process, two steps are performed:

Step 1: Set the encryption key, including the initial value x_0 and control parameter r, *for example* x= *0.123456789555777*, r=*3.9*.

Step 2: Do 150 times iteration using formula[1] (x_1.....x_{101}). In each iteration one floating point number will be generated. Each number consists of 15 digits after the decimal point.

3.2. Encryption Process of IPM

Since **prev_intra4x4_pred_mode_flag** and **prev_intra8x8_pred_mode_flag** syntax elements are control parameters, they can't be encrypted because that will introduce incorrect encoding results. SE **intra_chroma_pred_mode** also will not be encrypted to simplify the encryption process. **rem_intra4x4_pred_mode** and **rem_intra8x8_pred_mode** are holding the values of the selected best modes, thus they will be selected for the encryption process. Next is the pseudocode of the encryption process:

```
If (Up_Macroblock!=NULL && Left_Macroblock!=NULL)
{x = r* x * (1.0-x)
dec= x * 100
rnd=dec%8
current_IPM=XOR(current_IPM,rnd)}
```

First, the availability of neighboring macroblock is checked, to ensure the validity of new ciphered prediction modes. Next the random floating point number (chaotic number) is obtained by using expression in (1) and it is multiplied by 100. Mod 8 of this number is XORed with the original IPM to get the encrypted IPM.

3.3. Encryption Process of MVD

To reduce the time and computational complexity only the sign of MVD will be encrypted. If the random value generated using formula (1) is greater than 0.5, then the sign of MVD will be flipped else it will remain the same.

```
x = r * x * (1.0-x)
if (x>0.5)
        rnd=1
else rnd=0
mv_sign=XOR(mv_sign, rnd)
```

3.4. Encryption Process of Residue Coefficients

The encryption process for residue coefficients is done based on three stage as follow:

Stage 1:

Encrypting the sign of coefficients. if the random number by formula (1) is greater 0.5 then set the value of SE **coeff_sign_flag** as 1, else set as 0.

Stage 2:

In this stage, we encrypt only the suffix part of the bin-string of residue coefficients to avoid format incompatibility and increasing in bitrate. The suffix part is XORed with a random binary string of same length.

Stage 3:

To gain more ambiguity, the first DC coefficient in each block is encrypted. Fig. 1 shows the flowchart of encryption process for the first DC coefficient and next is the procedure:

1- Set *sign*=the sign of DC coeff.
2- Set *val*= the absolute value of DC coeff.
3- Set *len*=length of binary representation of DC coeff using mathematical formula (2)
4- Generate random sequence of length len-1 as follow:

```
for(n=0;n<len-1;n++){
        Set x using formula[1]
        fot=x*10;
        dd=mod(fot,2);
        jj=Shift_left(jj,1);
        jj=or(jj,dd);}
```

jj is the random number of length *len*-1 (Not of length len; to keep format compatibility and not increase the bitrate).

5- Set *nCoeff*= XOR (*val*, the random number*jj*).

6- Set the DC coeff= sign*$nCoeff$.

$$y = ceil\left(log_2(x+1)\right),\ where\ x\ is\ unsigned\ integer \quad (2)$$

4. EXPERIMENTAL ANALYSIS

For analyzing the results of the proposed encryption technique, we used H.264/14496-10 AVC reference software JM 19 and raw video sequences in QCIF format. The video sequences are: *"Bus", "City", "Crew", "Football", "Foreman", "Harbour", "Mobile", "Ice", and "Soccer"*. Each of them represents different combinations of motions, colors, contrast and objects. The first 100 frames in each sequence are selected for experiments. The results are compared with the techniques in [1] and [8] in section I.

4.1. Perception Security

Whenever it is possible to conceal the information of the plain-text form the human perception, it increases the protection against the interference of unauthorized users. Fig. 2 shows the encryption results using the proposed technique for the different video samples. The results shows that the encrypted samples are very chaotic and hardly to understood. Table.1 presents the PSNR values for measuring the quality of the encrypted video compared to the original one in the three color planes (Y U V). As we see, the PSNR values in most of the selected samples are somewhat small which indicates that the encrypted signal is highly opaque. In same table, the results of our proposed technique are compared with Selected Encryption(SE) technique in [1]. The results show that our proposed technique gives lower readings of PSNR compared with [1] in all the three color planes. Table 2 gives a comparison between the proposed technique and encryption technique in [8] for PSNR measurements. Figure 3 shows a comparison of the first encrypted frame of foreman video between the our proposed technique and SE technique in [1] at different QP values. As it is shown, the proposed technique is hiding the information of the picture more efficiently than [1], where in [1] the texture and facial features of the picture can be easily recognized. Table 3 shows the values of PSNR of foreman video at different QP values and compared with the values presented in [1]. It is shown that the proposed technique has less PSNR values which means that it more efficiently conceal the information of the video.

4.2. Cryptographic Security

Cryptographic security depends on the ciphers adopted by the encryption scheme. In the proposed technique, the adopted cipher is based on chaotic logistic map where its security analysis is clearly proved in [6].

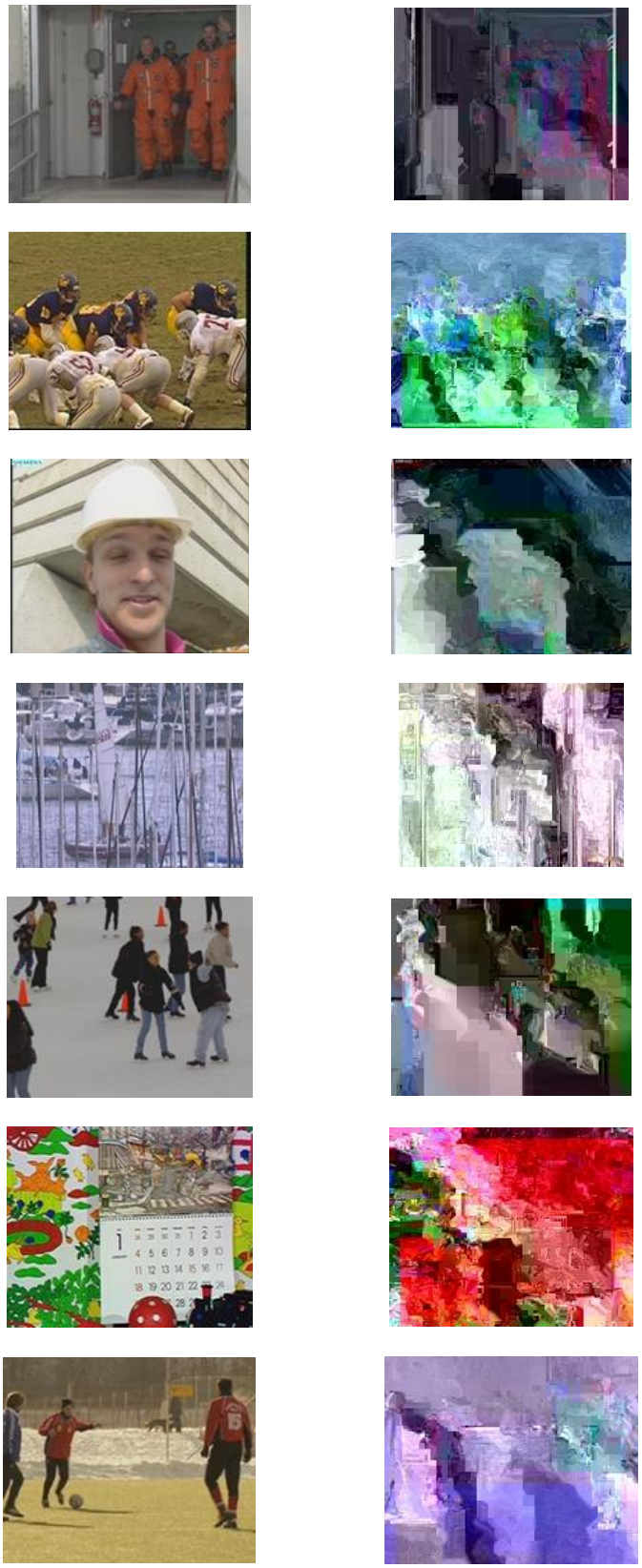

Figure 2. Encrypted Frames Using the Proposed Technique

4.2.1. Key Space

In the proposed technique, the secrete key is combination of the initial parameters of logistic map: the initial condition x_n and the control parameter r, where $x_n$is any floating point number [0,1) and r [3.5,4). Since $x_n$is double-precision floating point number then 64 bits are used to present this number in binary. Thus, the key space at least 2^{64} combinations to decrypt the cipher-text. In this case, Brute-Force Attack fails to decrypt the interested plaintext.

4.2.2. Replacement Attack

One of the common attacks in video encryption area is the replacement attack. This type of attacks is trying all the possible combinations of the parameters (Intra-Prediction modes, MVD and residue coefficients) that are used in the encryption process, to reveal more information about the cipher-text. Accordingly, we will count the number of trials that the attacker can do to get more clear decrypted frames. To break on macroblock, The attacker needs to break 16 sub-block of 4x4 size. Thus to guess one sub-block he needs$2^3 * 2^3 * 2^2 * 2 * P(16,16)$ trials to decrypt one 4x4 sub block. To decrypt the entire macroblock he needs to multiply the number by 16. Thus, to attack one macroblock using replacement attack, a huge number of trials need to be applied, so decrypting the entire frame will be almost impossible.

Table 1. PSNR Values of the Encrypted Videos Using Proposed Technique Compared With SE in [1]

	PSNR(Y)				PSNR(U)				PSNR(V)			
	Proposed		SE[1]		Proposed		SE[1]		Proposed		SE[1]	
	Orig.	Enc.	Orig.	Enc.	Orig.	Enc.	Orig.	Enc.	Orig.	Enc.	Orig.	Enc.
Bus	44.3	7.1	43.7	7.7	45.1	25.5	45.1	25.4	46.4	26.5	46.4	27.0
City	44.1	11.3	43.8	11.1	46.5	29.7	45.7	30.2	47.4	31.7	46.8	31.7
Crew	44.9	9.6	44.5	10.0	45.5	21.1	45.8	22.0	45.3	17.8	45.7	20.2
Football	44.8	10.4	44.2	11.3	46.0	16.0	45.7	14.6	46.3	22.3	46.1	24.3
Foreman	45.0	8.1	43.9	10.4	46.2	22.8	45.5	23.9	47.7	22.5	47.6	23.2
Harbour	44.1	9.8	43.7	9.8	45.1	20.7	45.4	22.9	45.9	31.0	46.6	31.7
Ice	47.2	9.9	46.1	10.4	48.8	25.3	48.6	25.3	49.2	19.4	49.1	19.7
Mobile	44.3	8.5	43.8	8.8	44.3	14.4	44.2	12.5	44.3	11.4	44.1	11.8
Soccer	44.9	10.4	43.6	10.6	46.8	17.4	46.5	20.8	47.8	21.8	47.8	22.2

Table 2. Comparison of PSNR Between the Proposed Technique Compared and Technique in [8]

	PSNR(Y)			
	Proposed		Technique[8]	
	Orig.	Enc.	Orig.	Enc.
Salesman	44.8	6.4	36.2	9.5
Foreman	45.0	8.1	36.7	11.4
Mobile	44.3	8.5	33.7	7.4

Table 3. Comparison of PSNR Between then Proposed Technique and SE in [1] For ForemanVideos in Different QP Values

	PSNR(Y)		PSNR(U)		PSNR(V)	
	Proposed	SE	Proposed	SE	Proposed	SE
18	8.1	10.4	22.8	23.9	22.6	23.2
24	8.9	9.7	22.1	24.9	24.3	25.0
30	8.9	9.2	22.8	24.9	24.1	24.0
36	8.0	8.2	23.7	24.3	23.0	23.3
42	7.3	8.6	24.9	26.8	25.0	24.6

Figure 3. Comparison of Encrypted Foreman Frame Between the Proposed Technique and SE in [1]

4.3. Computational Cost

4.3.1. Computational Complexity

The proposed technique is based on chaotic systems which are very simple and consumes very small computational power. In [1] the encryption process is based on AES technique which is very complex and needs much computational power. In [8], technique is based on encrypting the sign of levels in CAVLC but unfortunately this is increasing the bitrate of the stream.

Table 4. Rate of Increase in Processing Time in The Proposed Technique and SE[1]

Sequence	Proposed %	SE[1] %
Bus	0.056	0.25
City	0.038	0.23
Crew	0.121	0.14
Football	0.052	0.18
Foreman	0.032	0.20
Harbour	0.028	0.26
Ice	0.052	0.17
Mobile	0.026	0.33
Soccer	0.110	0.18

Table 5. Rate of Increase in Processing Time in The Proposed Technique and Technique [8]

Sequence	Proposed %	Technique[8] %
Foreman	0.032	0.9
Akiyo	0.170	1.1
Mother	0.058	0.7
Silent	0.155	1.0
News	0.245	0.5
Salesman	0.056	0.5

4.3.2. Computational Time

Table 4 gives the rate of the increasing of the computational time due to the encryption process. Since the computer at each test gives different execution time, the average of five runs is considered for each value in the table. It is observed, that the impact of encryption taxation on the encoding process is very small. The table also presents a comparison between the proposed technique and SE [1] for the increase of the computational time. It is clear that, the proposed technique consumes very small computational time compared

to SE technique in [1]. Table 5 gives another comparison between the proposed technique and the technique in [8]. The proposed technique has much less time impact compared to [8] for same types of video sequences.

5. CONCLUSION

This paper presents an efficient encryption technique to encrypt H.264/AVC video data. This algorithm is applied through the different modules of the compression pipeline which are: CABAC entropy coding, MVD, and residues transformation. The experimental results shows that the proposed technique is highly secured against different types of attacks. The proposed technique obtains high time efficiency compared to previous work algorithm, so it is not affecting the complete encoding process. In addition, the encrypted video is highly chaotic where the information can be completely opaque.

REFERENCES

[1] Zafar Shahid, Marc Chaumont, William Puech. "Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I and P frames," *IEEE Trans. Circuits Syst. Video Technol*. 2011; 21(5) May: 565-576.
[2] P.-C. Su, C.-W. Hsu, C.-Y. Wu. "A practical design of content protection for H.264/AVC compressed videos by selective encryption and fingerprinting," *Multimedia Tools Applicat*. 2011; 52(2–3) Jan: 529–549.
[3] S.-K. A. Yeung, S. Zhu, and B. Zeng, "Partial video encryption based on alternating transforms," *IEEE Signal Process. Lett.*, 2009; 16(10) Oct: 893–896.
[4] D. Marpe, H. Schwarz, T. Wiegand. "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol*., 2003; 13(7) Jul: 620–636.
[5] M. Podesser, H. Schmidt, and A. Uhl, "*Selective bitplane encryption for secure transmission of image data in mobile environments,"* In CD-ROM Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG 2002), Tromso-Trondheim, Norway, October 2002.
[6] S. Lian, J. Sun, Z. Wang, et al., "Security analysis of a chaos-based image encryption algorithm," *Physica A: Statistical* Mechanics *and its Applications*, 2005; 351(2–4): 645–661,.
[7] ITU-T Recommendation H.264. Series H: Audiovisual And Multimedia Systems. Infrastructure of audiovisual services - Coding of moving video. Advanced video coding for generic audiovisual services. 2012.
[8] S. Lian, Z. Liu, Z. Ren, H. Wang. "Secure advanced video coding based on selective encryption algorithms," *IEEE Trans. Consumer Electron.*, 2006; 52(2) May: 621–629.
[9] Y. Liu, C. Yuan, Y. Zhong. "*A new digital rights management system in mobile applications using H.264 encryption,"* in Proc. 9th Int. Conf. Adv. Commun. Technol., 2007; (1) Feb: 583–586.
[10] L. Tong, F. Dai, Y. Zhang, J. Li. "Prediction restricted H.264/AVC video scrambling for privacy protection," *Electron. Lett*., 2010; 46(1) Jan: 47–49.
[11] Y. Li, L. Liang, Z. Su, J. Jiang. "*A new video encryption algorithm for H.264*," in Proc. 5th ICICS, Dec. 2005: 1121–1124.
[12] E. Magli, M. Grangetto, G. Olmo. "*Conditional access to H.264/AVC video with drift control*," in Proc. IEEE ICME, Jul. 2006: 1353–1356.
[13] Y. Wang, M. O'Neill, F. Kurugollu. "*The Improved sign bit Encryption of Motion Vectors for H.264/AVC* ", EURASIP, 20th European Signal Processing Conference (EUSIPCO 2012), Bucharest, Romania, August 27 - 31, 2012: 1752-1756.
[14] S. Lian, J. Sun, G. Liu, Z. Wang. "Efficient video encryption scheme based on advanced video coding," *Multimedia* Tools *Applicat*., 2008; 38(1) Mar: 75–89.
[15] N.Bodke, J.Khule, P. Shinde, S.Kapse, K. Kumavat. "A Novel Approach for Codeword Substitution using Encrypted H.264/AVC Video Streams for Data Hiding" *International Journal of Computer Applications*, 2015; 128(4) October: 6-10.