Journal of Theoretical and Applied Information Technology
31st October 2018. Vol.96. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FAST SELECTIVE ENCRYPTION FOR VIDEO STREAM OF HIGH EFFICIENCY VIDEO CODING STANDARD

MOHAMMED A. SALEH, NOORITAWATI MD. TAHIR, AND HABIBAH HASHIM
Faculty of Electrical Engineering
Universiti Teknologi MARA(UiTM)
40450, Shah Alam, Selangor, Malaysia
mohamedoa@uitm.edu.my

ABSTRACT

In this paper, a lightweight encryption method is developed to protect the visual information of video. Due to the new video coding standards High Efficiency Video Coding (HEVC), and in order to meet the compression requirements a new enhancement for the existing encryption schemes is needed. This is because the lack of the ability of the current encryption methods to balance the security level with the compression efficiency of HEVC standard. Hence, we utilized the selective encryption for the sensitive syntax elements to secure video transmission and fulfill the attributes of HEVC standard. This is done via selection of sensitive syntax elements (Absolute Coefficient Levels) that are encrypted using Advanced Encryption Standard (AES) algorithm. On several video benchmarks, the analysis results obtained that the relevant security level of visual video perception is achieved, a trade-off between encryption reliability, flexibility, maintaining the HEVC bitstream format compliance, and computational complexity are provided. The protection level of this method is strong enough against plaintext and brute force attacks.

Keywords: *HEVC/H.265, selective encryption, video security, AES, bin string encryption.*

1. INTRODUCTION

Concurrently with the recent rapid increase in the use of internet and telecommunication technologies, security attacks have been increased due to the rapid growth in video sharing applications via smart devices. On the other hand, end users' consciousness about the confidentiality of their sensitive information that can be shared through the enormous internet network. Cisco has forecasted that mobile video sharing will occupy more than 69% of smart mobile data traffic for the year 2019, with total global percentage of smart mobile traffic targeted to be at 97% [1].

Video compression technique is used to maintain the bandwidth requirements by minimizing the data size that represents the video information. It is started as the popular analog video phone system in the year 1960 up to the recent High Efficiency Video Coding (HEVC) which was introduced by ITU-T and ISO/IEC in 2013 [2]. As we are aware, video consist of large data size, therefore, to ensure the security of data transmission over internet network, the video streaming requirements have to be taken into account, such as data communications, data retrieval, video contents compression and hardware resource. Moreover, since a new video compression standard HEVC was published lately, there is a demand for video streaming protection [3], [4]. Hence, in this study, we deemed further to design and develop an encryption method to secure HEVC video data.

There are two types of video data encryption techniques namely full encryption and selective encryption [5]. Due to high computational overhead generated by full encryption, only the selective encryption for video data will be considered here.

This method was proposed to secure all visual video information using Advanced Encryption Standard (AES). Hence the Context Adaptive Binary Arithmetic Coding (CABAC) in HEVC has been adopted to transform a video codec into a crypto coding module to achieve the encryption/decryption combined with encoding/decoding at the same time. In ensuring the robustness of encryption technique, sensitive syntax elements are selected for encryption purpose, while maintaining format compliance of video bitstream; the bit rate of the video stream; and computational complexity.

This paper is organized as follows; a brief background description about the HEVC entropy coding is presented in Section 2. Next, in Section 3, the previous researches related to HEVC video encryption approaches are discussed. Further, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

proposed selective encryption scheme is elaborated in Section 4. Then, the experimental implementation and results are presented and summarized in Sections 5. Finally, the conclusion of this paper is presented in Section 6.

2. HEVC ENTROPY CODING

The entropy coding techniques in the modern coding standards are Context Adaptive Variable Length Coding (CAVLC) and Context-Based Adaptive Binary Arithmetic Coding (CABAC) [6]. ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) have developed the CABAC algorithm within the joint H.264/AVC standardization process. During the ITU-T VCEG meeting that was held earlier in 2001, the new entropy coding method CABAC was introduced as a standard contribution and was denoted as the first version [7]. CABAC was implemented as one of the two alternative approaches for entropy coding of the H.264/AVC standard [8], [9]. However, Context Adaptive Variable Length Coding (CAVLC) was specified in H.264/AVC as a low-complexity entropy coding method, which depends on usage adaptively switched sets of variable length codes [10]. CAVLC encodes with low implementation complexity and low encoding efficiency as compared to CABAC [11], [8], [12].

In HEVC, the CABAC is also part of the first test model HM1.0 (HEVC reference software) that is combined with the Low Complexity Entropy Coding (LCEC) as a subsequent of CAVLC [12], [10]. Lastly, after the HEVC standardization improvements, the latest versions of HMs include CABAC only.

In CABAC, bin string statistical properties are utilized to compress video data, thus, it is denoted as a lossless compression scheme [13]. The entropy coding represents the video syntax elements as codeword or bit string, where the number of bits of that codeword is logarithmically proportional to the probability occurrence of the syntax element [14]. In HEVC standard, the syntax elements describe each of the properties of the Coding Tree Unit (CTU), Prediction Unit (PU), and Transform Unit (TU). The related syntax elements of the CTU are specified as follow: to describe the parameters for the block partitioning of the CTU into coding units (CU), to determine whether the CU is spatially predicted (Intra-frame) or temporally predicted (Inter-frame) and to describe the quantization parameters of the CU [15]. In the syntax elements of PU, the intra prediction mode, and the motion data are described. In the TU, the syntax elements describe the sign and magnitude of the quantized transform coefficients and non-zero coefficient map in TU.

The basic design of CABAC is a combination of three main phases that are: binarization; context modeling; and binary arithmetic coding. In Figure 1, CABAC encoding block diagram describes that phases as the main algorithmic building blocks. In binarization phase, the non-binary syntax elements are binarized into binary symbols (bins). In context modeling (regular coded), the probabilities of each bin are estimated based on some specific context. In the last stage (binary arithmetic coding), the bins compress to bits according to that estimated probability.

The proposed method in this work is based on the binarization stage in CABAC that responsible for formatting the non-binarized syntax elements into sequences of binary symbols called bins, which can also interpret in terms of a binary code tree. The binarization process is performed by different binarization methods namely Unary, Truncated Unary (TrU), kth order Truncated Rice (TRk), kth order Exp-Golomb code (EGk) or Fixed Length (FL) code binarization [9]. All of these methods are inherited from the binarization technique of H.264/AVC standard except the kth order Truncated Rice (TRk) method. The main idea of the binarization methods is how to represent the non-binary value N efficiently in a lesser number of bits.

The non-binary syntax elements in the HEVC can be binarized with the single binarization method from one of the five binarization techniques listed above or by combining more than one method, for instance, the *rem_Intra_Luma_pred_mode* element is binarized by FL while *coeff_abs_level_remaining* (*calr*) is binarized by TrU, TRk, and EGk [10].

3. PREVIOUS PROPOSED SELECTIVE ENCRYPTION METHODS ON HEVC

In order to reduce the computational overhead of the data encryption, the selective encryption is used. Whereas the determined parts of the compressed bit stream are encrypted to provide sufficient security [13]. There are different types of encryption algorithms that can be used for selective encryption for video or image data, for example, AES and Data Encryption Standard (DES) [16], [4]. It was found that previous research work on video encryption applied the encryption algorithms on the video data at different stages for instance prior to entropy coding or before transform coding.

Since HEVC standard has been developed recently, there are a limited number of the proposed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

encryption methods. Firstly, in 2014 Hofbauer et. al. [17] proposed a scheme on HEVC standard based on encrypting the sign bits in the luminance channel only. The limitation of this method is as reported in [3], which stated, the encryption for the sign bits is not able to provide higher security level. Later in 2014, Shahid [18] proposed a new selective encryption method for HEVC standard on the CABAC entropy codec. In his method, some of the syntax elements are selected and encrypted using AES where these elements consist of the sign bit of Quantized Transform Coefficient (QTCs), the suffix of TRp code, the suffix of EG0 code, the sign of MVDs and the suffix of EG1 code. The advantage of this method is that, it is able to provide highly efficient security. However, due to the percentage of the encrypted data, the computational complexity is increased. Furthermore, the bit rate is also increased by the padding function that was applied on the plaintext. Another relative encryption scheme was proposed by Memos in [19]. The author proposed to encrypt all video data in I frame by AES algorithm. Since the higher percentage of video information is in I frame, efficient security level is provided but in return it generates an increase in the computational complexity and bitrate. In [20], Saleh et al. proposed a new encryption approach to secure moving information in the video by encrypting the Motion Vector Difference (MVD) using AES.

4. THE PROPOSED SELECTIVE ENCRYPTION FOR HEVC

As mentioned earlier, the latest version of HEVC standard only supports CABAC in all coding configurations. To avoid computational complexity, only some of bypassed syntax elements are selected as encrypted segment, Figure 1 shows the encrypted part of bypassed syntax elements in CABAC. The encryption process has been done on the selective syntax elements using AES, which denoted as the most robustness encryption standard. AES offers five encryption modes namely CBC, ECB, CTR, OCB, and CFB whilst Cipher Feedback (CFB) mode is known as the most suitable mode for streaming encryption [5]. Thus, the CFB mode has been used to encrypt the specific syntax elements. In this approach, entropy coding stage serves the purpose of encryption syntax elements without affecting the coding efficiency of video coding standard by completely maintaining the compliant of a bitstream with low computational power.

The selective syntax elements can be produced from the non-zero coefficients in the transform unit. However, the transform unit which does not contain a non-zero coefficient whilst the zero coefficients in non-zero transform unit is not encoded. In the entropy coding stage, the sensitive syntax elements can be selected from CABAC as an important effective factor. As explained earlier, every non binary syntax element can be binarized by one or more method of the five binarization technique. Table 1 shows the main syntax elements and their binarization techniques with selected element highlighted. Recall that *coeff_abs_level_remaining*, is the remaining absolute value of a transform coefficient level that is coded with Golomb-Rice code at the scanning position *n* and if *coeff_abs_level_remaining* is not present, it is assumed to be nil [19] .

As for the CABAC coding process, the bins can be either regular (context) coded or bypass coded. In bypass coded, bins do not require regular encoding which allows these bins to be processed at a much high throughput than context coded bins [15], [20], [20].

Further, the binarization for the absolute coefficient levels is based on concatenated application of three binarization processes specifically truncated unary (TrU), kth order truncated Rice (TRk) and kth order Exp-Golomb (EGk).

Table 1: Main Syntax Elements and Their Binarization Process

Syntax Structure	Syntax Element	Binarization
slice_segment_data	All	FL
sao	All	FL, TrU
coding_quadtree	All	FL
coding_unit	rem_Intra_Luma _pred_mod	FL
	Others	FL, TrU
prediction_unit	All	FL, TrU
transform_tree	All	FL
mvd_coding	abs_mvd_minus2	EG1
	Others	FL
transform_unit	All	FL
residual_coding	*coeff_abs_level_remaining*	TRk, TrU, EGk
	Others	FL, TrU

4.1 Selection and Encryption for Syntax Elements (Plaintext)

Since the *calr* represents the remaining absolute value of the transform coefficient levels for every transform unit (TU), the TU is split into coded sub-blocks with each of them is a 4×4 pixel block and each of these blocks is scanned and the absolute coefficient levels syntax elements are generated.

Absolute coefficient levels are classified as three main syntax elements specifically:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1. Coefficient absolute levels greater than one *coeff_abs_level_greater1* (*calg1*);

2. Coefficient absolute levels greater than two *coeff_abs_level_greater2* (*calg2*);

3. Coefficient absolute levels remaining *coeff_abs_level_remaining* (*calr*).

The first and second syntax elements are represented by one bit for each other while the *calr* is represented by a series of bits. Due to the length of calg1 and *calg2* (one bit), the encryption is useless [3]. In this approach, the sequences of a specific number of *calr* syntax elements are denoted as the plaintext stream input of the encryption algorithm P_i. Further, Figure 2 illustrated the encryption process in the proposed method using AES algorithm with the CFB mode for *calr* that will be encrypted if it is binarized using Exp-Golomb and if the number of Transform Coefficient Count (TCC) (*calr* syntax element in the TU) is more than four instead of each syntax elements. TCC represents the number of the non-zero coefficient value in any TU. This condition has been selected because the value four was found as an optimal value of the TCC, where if the value of TCC equal to or less than four gives a non encrypted video while if the TCC greater than four gives an encrypted video.

4.2 Decryption Process

At the decoder end upon completion of the arithmetic decoding process, the original plain text P_i (*calr*) is retrieved (decrypted) from the ciphertext C_i using the proposed algorithm and same encryption key. Note that the original plain text can be retrieved by applying the same procedure that is used for generating the cipher text. In other words, to substitute and generate the original syntax element of the absolute coefficient level, similar encryption key Ek of AES is used to generate the keys stream.

5. EXPERIMENTAL IMPLEMENTATION AND RESULTS

Using the HEVC test model HM10, the simulation is performed on a system as described in Table 2. All experimental and analysis are performed on several types of benchmark video sequences. The utilized video classes in terms of resolution and frame rate of each video sequences are tabulated in Table 3. The experimental results performed by encryption and decryption *calr* of the HEVC in case the number of these elements in the transform unit are more than four. The used coding configuration of HEVC is Low Delay Main.

The visual distortion for the encrypted videos is conducted and the results are as observed in Figure 3 for *BasketBallDrill* video sequences. Furthermore, the bit rate and total data of encrypted and non-encrypted for various type of benchmark video sequences for quantization parameters 32 to 36 using Low Delay Main coding configuration are described in Table 4.

Table 2: Experiment PC Properties

Experimental setup	
Processor	Intel(R) core(TM) i5, CPU 3.00GHZ
RAM	8.00GB
Number of Frames Encoded	100
HEVC Test Model	HM10
Frame Rate	Varied According to the video sequence
Coding Configurations	Low Delay
Quantization parameter (QP)	32 - 36

Table 3: The Set of Benchmark Video Sequences Used for Simulate the Video Encryption Method on HEVC

Class	Sequence	Resolution	Frame Rate
A	Traffic	2560×1600	30
	PeopleOnStreet	2560×1600	30
B	ParkScene	1920×1080	24
	Kimono	1920×1080	24
C	BasketBallDrill	832×480	50
	BQMall	832×480	60
	PartyScene	832×480	50
	RaceHorseC	832×480	30
D	BasketBallPass	416×240	50
	BQSquare	416×240	60
	BlowingBubbles	416×240	50
	RaceHorses	416×240	30
E	Vidyo1	1280×720	60
	Vidyo3	1280×720	60
	Vidyo4	1280×720	60

Table 4: Bitrate and Data Size for Non-Encrypted and Encrypted Video Sequences

Sequence	Bitrate(kbps)		Total size	
	Non-Encry.	***Encrypted***	***Non-Encry.***	***Encrypted***
Traffic	5019.456	5113.320	209144	213055
PeopleOnStreet	11112.792	11183.208	463033	465967
ParkScene	2493.888	2540.294	129890	132307
Kimono	2111.174	2174.266	109957	113243
BasketBallDrill	1128.200	1139.600	28205	28490
BQMall	1738.176	1744.608	36212	36346
PartyScene	3783.760	3827.200	94594	95680
RaceHorseC	1647.672	1655.160	68653	68965
BasketBallPass	343.480	346.760	8587	8669
BQSquare	828.192	858.864	17254	17893
BlowingBubbles	573.680	570.680	14342	14267
RaceHorses	513.144	517.560	21381	21565
Vidyo1	1048.368	1057.296	21841	22027
Vidyo3	1237.584	1286.832	25783	26809
Vidyo4	900.624	907.056	18763	18897

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5.1 Video Quality Analysis

In order to evaluate the encryption performance of this proposed method, the video quality is tested using two video quality performance measures, Peak Signal-to-Noise Ratio (PSNR) [21] and Structural Similarity Index Metric (SSIM) [22]. Both PSNR and SSIM are used for evaluating and validating the quality of the original video as compared to the encrypted version of these videos.

Furthermore, Table 5 tabulates the comparison of analysis metric PSNR and SSIM of all encrypted and non-encrypted benchmark video sequences. The average value of PSNR for all non-encrypted benchmark video sequences is 34.75 dB whilst it is 8.4 dB for all encrypted benchmark video sequences. In addition to that, the results of SSIM upon evaluation of the quality difference between encrypted and non-encrypted benchmark video sequences are described, where the SSIM average value for the original encoded video is 0.92 dB and 0.30 dB for the encrypted video.

From the analysis results of PSNR and SSIM, it is proven that the proposed method is strong enough for distorting the video contents information (perceptual encryption) and suitable for each video sequences classes.

5.2 Computational Analysis

In any encryption method, the computational overhead and bit rate need to be taken into consideration. Here, Table 6 tabulate the encryption and decryption time as compared to the encoding and decoding time without encryption based on different benchmark video sequences. Also in Table 7, the time of encryption and decryption BasketBallPass sequence with different encoded frames is also tabulated. It is observed from the results that the delay is acceptable to use this method for the video streaming because the percentage of the encrypted data is low as compared to the overall video data, which directly proportion with the computational complexity and time delay, thus the generated computational complexity is low. The effecting of the encryption method on the encoding processor in terms of computational overhead is described in Table 8, where we have calculated the impact of the encryption process on the computational complexity using PC as detail in Table 2. The computational weight requirements are acceptable.

Figure 4 (last page) shows the impact of the encryption process on encoding and decoding time for *BasketBallPass* video sequence.

Table 5: PSNR and SSIM Comparison for Encrypted and non-Encrypted Video Sequences

Sequence	PSNR (dB)		SSIM (dB)	
	Original	Encrypted	Original	Encrypted
Traffic	36.19	5.79	0.940	0.275
PeopleOnStreet	34.51	7.18	0.920	0.280
ParkScene	34.77	5.34	0.900	0.194
Kimono	37.45	6.26	0.93	0.34
BasketBallDrill	34.82	7.76	0.890	0.277
BQMall	34.17	9.27	0.940	0.552
PartyScene	31.10	8.61	0.920	0.145
RaceHorseC	32.39	17.24	0.910	0.431
BasketBallPass	34.50	8.56	0.910	0.178
BQSquare	31.58	6.43	0.900	0.107
BlowingBubbles	32.06	10.28	0.900	0.293
RaceHorses	32.06	9.09	0.907	0.174
Vidyo1	38.87	6.44	0.950	0.360
Vidyo3	37.97	9.96	0.950	0.523
Vidyo4	38.82	7.80	0.950	0.426
Average	34.75	8.40	0.92	0.30

Table 6: Processing Time of Encoding and Decoding with and without Encryption for All Benchmark Video Sequences

Video Sequences	Time (Second)			
	Non-Encrypted		Encrypted	
	Encode	Decode	Encode	Decode
Traffic	5003.947	18.085	5274.270	13.020
PeopleOnStreet	8470.463	32.024	10423.230	21.651
ParkScene	2738.561	10.395	3283.287	10.563
Kimono	3954.820	14.428	4090.055	15.526
BasketBallDrill	656.453	2.652	692.055	3.644
BQMall	548.293	2.204	609.181	1.876
PartyScene	743.133	3.626	917.660	3.633
RaceHorseC	962.970	3.837	1155.304	3.975
BasketBallPass	432.587	1.347	478.655	1.351
BQSquare	137.044	0.715	171.183	0.717
BlowingBubbles	155.392	1.102	169.176	0.792
RaceHorses	237.220	1.045	307.861	1.035
Vidyo1	950.643	3.470	991.059	3.166
Vidyo3	1019.744	3.572	1061.568	2.874
Vidyo4	970.183	3.534	1006.974	3.512

Table 7: HEVC Encoding/Decoding Processing Time Comparison Encryption and without Encryption for the BasketBallPass Video Sequence

Frame	Encoding Time (s)		Decoding Time (s)	
	Original	Encryption	Original	Encryption
10	132.35	148.05	0.59	0.59
20	280.33	303.76	0.95	0.96
30	431.57	478.66	1.32	1.34
40	576.04	644.35	1.70	1.70
50	753.70	830.48	2.16	2.15
60	942.92	1041.57	2.65	2.67

Table 8: Analysis of CPU Processing for HEVC Encoder and Decoder with Encryption and without Encryption

Sequence	CPU% Usage	CPU% Usage

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Original	*Encrypted*	*Original*	*Encrypted*
BasketBallPass	19.38	19.66	0.09	0.07
BQSquare	21.74	17.62	0.11	0.06
BlowingBubbles	15.21	22.76	0.15	0.07
RaceHorses	19.93	19.08	0.12	0.35
Vidyo1	23.56	15.05	0.18	0.23
Vidyo3	14.48	22.12	0.12	0.44
Vidyo4	23.23	17.93	0.15	0.42

5.3 Security Analysis

In this section, the security of the encryption approach is analyzed and evaluated according to the following points:

- Entropy, and local standard deviation;
- Encrypted video correlation
- Effectiveness of an encryption key
- Protection against the Key Plaintext and Brute Force Attacks

5.3.1 Analysis of entropy, and local standard deviation

Here, the proposed approach has been tested by measuring the frame data contents with the entropy via comparing the original frames with the encrypted frames. This analysis is performed by utilizing the statistical randomness to characterize the texture of frame. Therefore, the quality of the frame is measured using the image entropy H(X). Recall that as stated by Shannon [23] the original frame has higher entropy value and lower standard deviation value than the encrypted frame. In addition to frame entropy analysis, the local standard deviation σ(j) of the noise and interference of video frame for the encrypted frame has been analyzed by comparing it with the original frame. The frame entropy is retrieved as in (1) while the local standard deviation σ(j) is computed using (2).

$$H(X) = -\sum_{i=0}^{2^k} p(\alpha_i) \log 2(\alpha_i). \quad (1)$$

$$\sigma(j) = \sqrt{\frac{1}{m}\sum_{i=1}^{m} p(i) - \overline{p(j)}} \quad (2)$$

where α_i is the frame gray levels, $p(\alpha_i)$ is the probability of gray level, *k* is the number of bit per pixels (i.e. in this test, *k* equal to 8), $\overline{P(j)}$ is the neighbor pixel's local mean, *m* is the pixel block size to calculate the local mean and standard deviation, and M is the size of the frame.

By applying equation (1) on the original frame of *Vidyo3*, the entropy is 6.9188 bits/pixel, where its value of the encrypted frame is 2.4184 bits/pixel. Next, using equation (2), the mean local standard deviation value for the original frame of *Vidyo3* is 59.1033 gray levels while the mean local standard deviation of the encrypted frame is 124.96 gray levels. These results demonstrated that the proposed method is robust for protecting the encrypted video against statistical attacks.

5.3.2 Encrypted video correlation

As we are aware, the pixels in the frame are normally correlated with each other. Therefore, in the original frame, the correlation between pixels is high. Whereas, if the encrypted frame has high correlation pixels that indicated the frame can be detected, that is to say, the encrypted information contents of a frame can lead to retrieving the original image easily. In this proposed approach, low correlation pixels within the encrypted frame are produced where the correlation of a pixel with its neighbouring pixel is specified as in (3).

$$corr(x,y) = \frac{1}{n-1}\sum_{0}^{n}\left(\frac{x_i - \overline{x_i}}{\sigma_x}\right)\left(\frac{y_i - \overline{y_i}}{\sigma_y}\right) \quad (3)$$

where, $\overline{x_l}$ represents the horizontal local mean, $\overline{y_l}$ is the vertical local mean, n is the number of pixels (x_i, y_i), σ_x and σ_y represent standard deviation of x and y respectively. The normal value of the pixels correlation is $corr(x,y) \cong 1$, the correlation of adjacent pixels in the original frame of *Kimono* video sequence is 0.9997 and it is equal to 0.1461 in encryption frames of the same sequence. As noted, the results clearly showed major difference between the correlation of original and encryption frames pixels. Thus, from these results, the robustness of the proposed encryption method has been proven.

5.3.3 Protection against the key plaintext and brute force attacks

Here the proposed scheme against the Key Plaintext Attack (KPA) is analyzed. KPA is the retrieving for encrypted data based on the non-encrypted data. The encrypted data can be easily predicted using the KPA if the encrypted data is a single bit such as sign bits of transform coefficient or sign bits of Motion Vector Difference (MVD). In this approach, the encrypted data length is more than one bit, hence, it gives a robust encryption against the brute force attack. Using AES algorithm to encrypt *calr* data, the ciphertext is not vulnerable to KPA as has reported in [24].

Furthermore, it has been established that deriving the key from the encrypted bits that were encrypted using AES, using KPA or a brute force attack is difficult and complex although from the original frame [25].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. CONCLUSION

In conclusion, enhancement method for video encryption is developed for High Efficiency Video Coding standard. This is done by encrypting the most significant syntax element in HEVC bit stream for the entropy coding of HEVC standard. Further, selective encryption of HEVC is performed using AES algorithm. The experimental results of the proposed scheme showed that the reliable perceptual security is attained along with the protection against both of the key plain text and brute force attacks while maintaining the HEVC bitstream format compliance with low computational complexity. Consequently, from the experiment results, it can be concluded that the proposed encryption method is suitable for the real-time application and limited resource systems.

ACKNOWLEDGMENT

The authors would like to thank the Ministry of Higher Education (MOHE) Malaysia for providing the grant 600-RMI/NRGS 5/3(5/2013), and Research Management Institute (RMI) of Universiti Teknologi Mara for supporting this research work.

REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019." [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html. [Accessed: 29-Jul-2017].

[2] G. J. Sullivan, J. Ohm et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012, pp. 1649–1668.

[3] Y. Wang, S. Member et al., "A Tunable Encryption Scheme and Analysis of Fast Selective Encryption for CAVLC and CABAC in H.264/AVC," *IEEE Transic. Circuits Syst. Video Technol.*, vol. 23, no. 9, Sep. 2013, pp. 1476–1490.

[4] S. Gupta, L. Kishor et al., "Comparative Analysis of Encrypted Video Streaming in Cloud Network," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, Jul. 2014, pp. 5470–5476.

[5] M. A. Saleh, N. Tahir et al., "An Analysis and Comparison for Popular Video Encryption Algorithms," in *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2015, no. 12–14, pp. 90–94.

[6] D. Marpe, H. Schwarz et al., "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, Jul. 2003, pp. 620–636.

[7] Iain E. Richardson, *The H.264 advanced video compression standard*, 2nd Editio. JohnWiley & Sons, 2010.

[8] D. Arpe, H. Schwarz et al., "Context-Based Adaptive Binary Arithmetic Coding," *IEEE Trans. circuits Syst. video Technol.*, vol. 13, no. 7, 2003, pp. 620–636.

[9] D. Marpe, SchWarz et al., "Method and Apparatus for Binarization and Arithmetic Coding of a Data Value," US 7,088,271 B2, 2006.

[10] G. J. S. Vivienne Sze, Madhukar Budagavi, *High Efficiency Video Coding (HEVC) Algorithms and Architectures*. Springer, 2014.

[11] T. Davies and A. Fuldseth, "JCTVC-F162: Entropy coding performance simulations," *Jt. Collab. Team Video Coding*, 2011.

[12] B. Peng, D. Ding et al., "A Hardware CABAC Encoder for HEVC," in *IEEE Symposium on Circuits and Systems (ISCAS)*, 2013, pp. 1372–1375.

[13] T. Lookabaugh and D. C. Sicker, "Selective Encryption for Consumer Applications," in *IEEE Conference Consumer Communications and Networking*, 2004, pp. 124–129.

[14] T. Wiegand, G. Sullivan et al., "Joint Draft ITU-T Rec. H.264 | ISO/IEC 14496-10 / Amd.3 Scalable video coding," *Jt. Video Team JVT-X201*, vol. 108, 2007, p. 563.

[15] V. Sze and M. Budagavi, "Reducing Context Coded and Bypass Coded Bins to Improve Context Adaptive Binary Arithmetic Coding (CABAC) Throughput," US 2013/0272389 A1, 2013.

[16] A. Pande, P. Mohapatra et al., "Securing Multimedia Content using Joint Compression and Encryption," *IEEE Multimed.*, vol. 20, no. 4, May 2012, pp. 50–61.

[17] H. Hofbauer, A. Uhl et al., "Transparent Encryption for HEVC Using Bit-Stream-Based Selective Coefficient Sign Encryption," in *IEEE Conference on Acoustics, Speech and Signal Processing*, 2014, no. 4–9 May, pp. 1986–1990.

[18] Z. Shahid and W. Puech, "Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings," *IEEE Trans. Multimed.*, vol. 16, no. 1, Jan. 2014, pp. 24–36.

[19] V. A. Memos and K. E. Psannis, "Encryption algorithm for efficient transmission of HEVC

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

media," *J. Real-Time Image Process.*, vol. 12, no. 2, 2016, pp. 473–482.

[20] M. A. Saleh, N. M. Tahir et al., "Moving Objects Encryption of High Efficiency Video Coding (HEVC) using AES Algorithm," *J. Telecommun. Electron. Comput. Eng.*, vol. 3, no. 2289–8131, 2016, pp. 31–36.

[21] C. A. Kim, Seung-Hwan. Kerofsky, Louis Joseph. Segall, "Golomb-Rice/EG Coding Technique for Cabac In HEVC," WO 2013/153820 Al, 2013.

[22] J. Lou and L. Wang, "Method of Determining Binary Codewords for Transform Coefficients," US 0202029 A1, 2013.

[23] A. Tanchenko, "Visual-PSNR measure of image quality," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, Jan. 2014, pp. 874–878.

[24] Y. A. Y. Al-najjar and D. C. Soong, "Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI," *Int. J. Sci. Eng. Res.*, vol. 3, no. 8, Aug. 2012, pp. 1–5.

[25] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, Oct. 1949, pp. 656--715.

[26] A. Unterweger, "Post-Compression Multimedia Security," Ph.D. thesis, Dept. Comp. Scin., University of Salzburg, Salzburg, Austria, 2014.

[27] A. Bogdanov, D. Khovratovich et al., "Biclique cryptanalysis of the full AES," in *Conference on the Theory and Application of Cryptology and Information Security*, 2011, pp. 344–371.

Journal of Theoretical and Applied Information Technology
31st October 2018. Vol.96. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

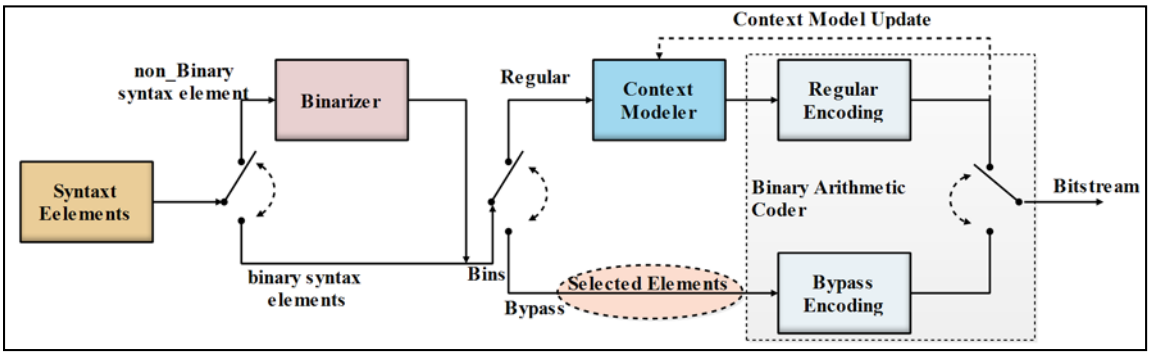

Figure 1. Block Diagram of CABAC

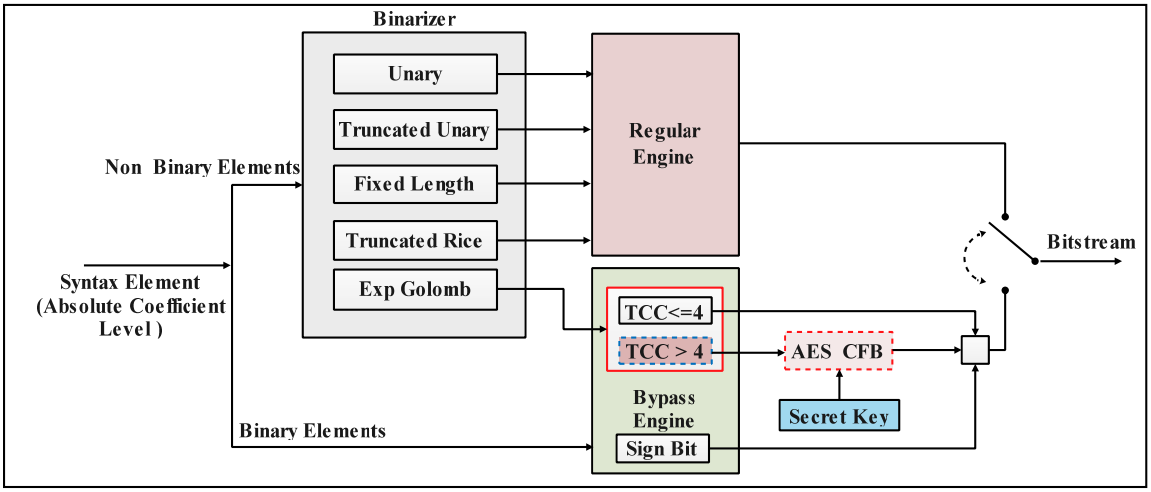

Figure 2. Encryption process of proposed method

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

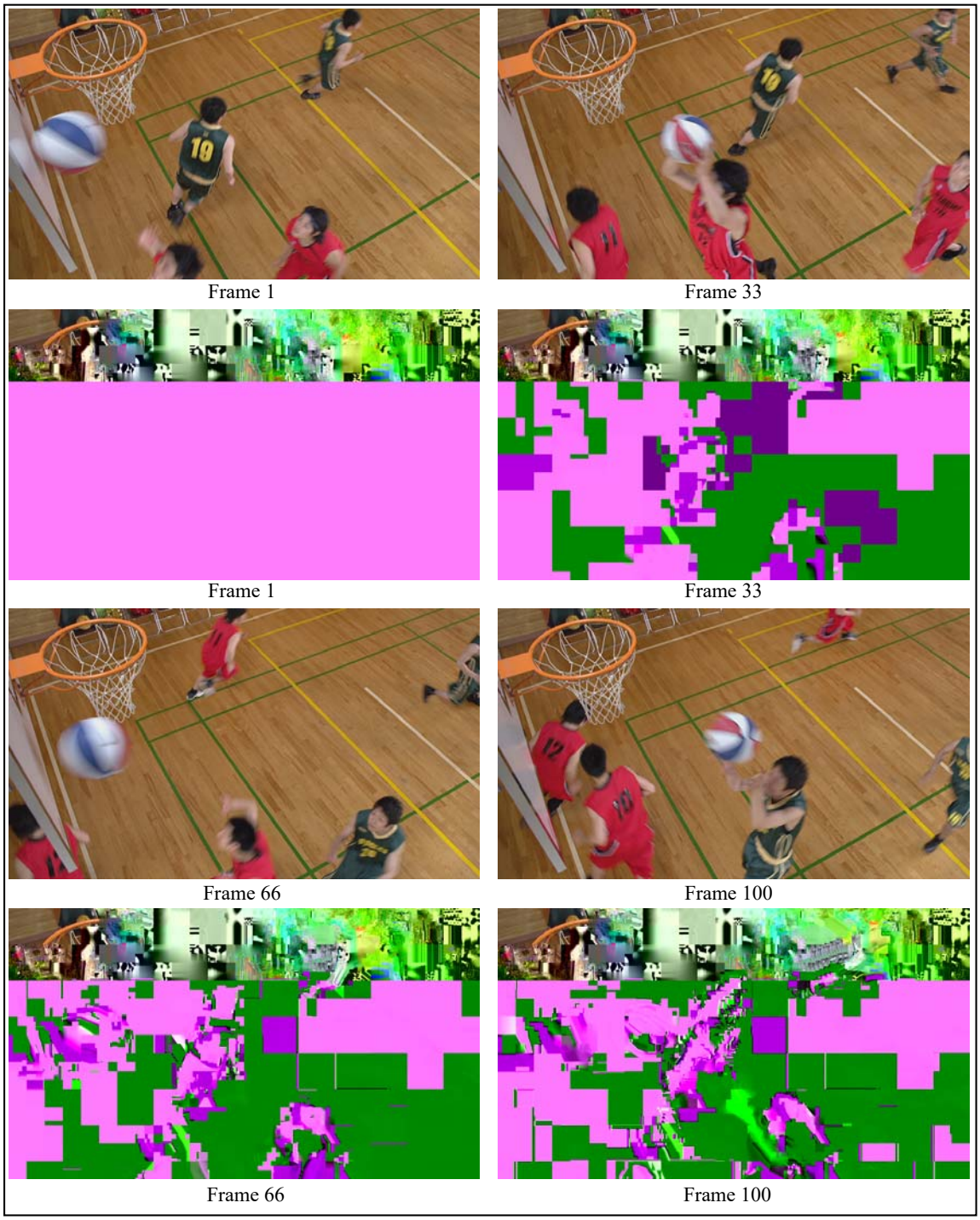

Figure 3: Encrypted video sequence of BasketBallDrill 832×480 using low delay coding configuration

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

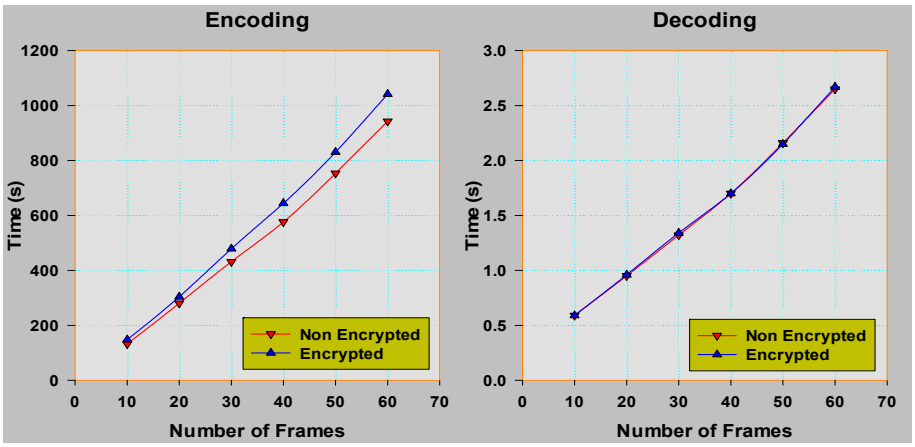

Figure 4: Time taken by encryption for Absolute Coefficient Level of BasketBallPass video sequence