Journal of Real-Time Image Processing manuscript No.
(will be inserted by the editor)

Encryption Algorithm for Efficient Transmission of HEVC Media

Vasileios A. Memos and Kostas E. Psannis

Received: date / Revised: date

Abstract Recently, H.265/MPEG-H or High Efficiency Video Coding (HEVC) as it is well known, has been established as better compression standard due to reduction of about 50% bit-rate for same video quality and less bandwidth consumption, compared to its predecessor H.264/MPEG-AVC standard. Many algorithms have been proposed and developed for efficient and secure streaming of multimedia files. However, these methods do not meet all the requirements which a media file should have to be transmitted effectively and securely over the internet. In this paper, we present a new encryption and transmission algorithm for efficient HEVC delivery. Experimental results demonstrate that our proposed algorithm is more secure and effective compare to previous algorithms used for H.264 standard and shows better overall performance.

Keywords H.264, HEVC, transmission, encryption, AES algorithm.

1 Introduction

H.265/MPEG-H or High Efficiency Video Coding, known as HEVC, is the latest compression standard, which was officially approved in January 2013 [1], and became the successor of H.264/MPEG-4 or AVC (Advanced Video Coding) standard [2]. The HEVC standard design has the features to be easily adaptable to about all the current existing H.264/MPEG-AVC applications, and emphasizes mainly on the capability of Ultra-High-Definition (UHD) video view [2] without much bandwidth consumption.

The basic achieve of the HEVC compression standard is the fact that it presents significantly better compression performance compared to the existing standards. Specifically, the HEVC standard causes about 50% bit-rate reduction for about the same video quality, compared to H.264/MPEG-AVC standard [1], [3], [27]. In addition, it is designed to provide high-quality streaming multimedia, even on low-bandwidth networks, due to the fact that it consumes about half bandwidth less than H.264/MPEG-AVC. Therefore, the use of HEVC compression standard brings many benefits against compared to its predecessor H.264/MPEG-4 Advanced Video Coding (AVC) standard [2], [4].

It is notable that HEVC standard presents specific complexity [4], [5], [28] and implementation [5], and it is being integrated into multimedia systems and protocols [6], while constitutes the current codec for resolutions beyond HDTV [7] for real-time streaming of video files [8]. Moreover, HEVC presents more effective rate-distortion (R-D) performance, by using specific algorithms [9].

Due to the above features which established HEVC as the best compression standard, several researches focus on the development of security methods which can contribute to the protection of HEVC videos, while they are transmitted over the internet. Specifically, special encryption algorithms have been proposed for HEVC standard to protect the video sequence against cryptanalysis attacks by malicious users who use third-party tools and methods to crack and steal the transmitted video sequence.

In this paper we present a new encryption algorithm for efficient secure transmission of video files compressed with HEVC standard. Our algorithm is based on known algorithms

V. A. Memos
Department of Applied Informatics,
University of Macedonia,
156 Egnatia Street, Thessaloniki 54006 , Greece.
E-mail: tm0844@uom.edu.gr

K. E. Psannis
Department of Applied Informatics,
University of Macedonia,
156 Egnatia Street, Thessaloniki 54006 , Greece
E-mail: kpsannis@uom.gr

proposed for previous compression standards, which we adapt properly so as to be applicable to the new standard.

The paper is organized as follows: In Section 2 we present the related work of other researchers on video encryption area, both on HEVC standard and previous compression standards. In Section 3 we present and analyze our proposed algorithm for efficient encryption and transmission video files, compressed with HEVC standard. Section 4 describes our methodology for the experiments we made upon the proposed algorithm, while Section 5 includes the experimental results with comparative diagrams. Section 6 concludes the paper and indicates future research directions.

2 Related Work

Multiple algorithms and schemes have been proposed for video encryption by many researchers, both in HEVC compression standard and in previous standards, such as H.264 and MPEG. Their main properties and limitations are presented in [10]. The authors make a series of comparisons to conclude that there is no method that can meet all the security requirements and thus, the suitable encryption algorithm for each video case depends on its confidentiality requirements.

A novel selective encryption scheme for secure transmission of video streams compressed with H.264/AVC standard is proposed in [11]. Simulation results demonstrate that its application implies PSNR degradations of about 25 to 30 dB when the ciphering key is unknown and thus, the video becomes unidentifiable. Another relative encryption algorithm especially for H.264/AVC format is proposed in [12] and its experimental results demonstrate much less important data encryption, better security and high efficiency. Other selective encryption algorithms for image and videos compression standards: JPEG, JPEG2000, H.264/AVC and H.265/HEVC are analyzed with cryptanalysis methods in [15].

In addition, encryption method and algorithm for Intra and Inter frames in MPEG videos are presented in [20]. According to the authors, highly private videos require encryption of all parts of the video, because all these are important in such cases. Thus, both Intra and Inter frames need to be encrypted. Another security scheme for MPEG video standard too, is proposed in [21] and is based on AES-128 encryption algorithm. The difference here is that the authors choose and encrypt only the Intra frames of the video, because Inter frames are useless without knowing the corresponding Intra frames. This process saves 30-50% of encryption/decryption time and does not affect the size of the encrypted stream.

A special study on applicability and encryption of H.264 video format including its Scalable Video Coding (SVC) extension, presented in [13]. This survey is based on the latest results on video encryption methods which have been proposed. A scalable video encryption algorithm for H.264/SVC too, is proposed in [14], which shows adequate performance and strength against cryptanalysis attacks, and it seems that it can be used in real-world applications. Moreover, possible bitstream elements, which can be used for HEVC compatible encryption, are described in [18], and ensure a good level of protection of the video information.

Finally, there are many encryption algorithms proposed exclusively for HEVC standard, based on selective encryption. Specifically, a new scheme based on selective encryption for HEVC was proposed in [16] and ensures transparent and sufficient encryption and protection against attacks. Moreover, this scheme allows fast encryption and decryption, while preserve the format and length of the video stream. A new scheme for format compliant visual protection of HEVC using selective encryption too, was proposed in [17], and offers a good level of protection with minimal use of computational requirements. Similar to this project, an efficient SE system for CABAC entropy coding of HEVC video standard was proposed in [19], which presents sufficient protection against cryptanalysis attacks, while makes it proper for streaming on heterogeneous networks, due to the fact that bit-rate remains the same and the system requirements are minimal.

Although the new algorithms which have been proposed exclusively for HEVC standard, present several advantages and are regarded to be effective solutions for protection the video sequence, we merge two known algorithms proposed for previous standards, [20], [21], and modify them properly so as to be integrated with HEVC standard, offering encryption and decryption time savings, while ensuring the protection level of the sequence against malicious users.

3 Proposed Algorithm

Our proposed algorithm for efficient encryption and secure transmission is based on Advanced Encryption Standard (AES), amendable to be adaptable to the new video compression standard, HEVC. AES was adopted by the U.S. government in 2002 and became the successor of the Data Encryption Standard (DES) algorithm which was launched in 1977 [24]. This algorithm, known as Rijndael too, is a symmetric-key algorithm, fact which means that the same key is used for both encrypting and decrypting video files. Figure 1 depicts the encryption process of a video, by using a 256-bit key to our proposed cryptographic algorithm, which is described below.

J. Nehete et al. proposed a real-time MPEG video encryption algorithm using AES with key length of 128 bits (AES-128) [21]. In this paper we adopt their proposed algorithm, making it amendable to AES-256 to ensure maximum security, due to the fact that larger key-sizes use is more secure. Table 1 shows the required time to crack an encryption algorithm of a specific key size by using brute force attack [25]. As it is clearly shown, the larger key sizes ensure more security, due to the fact that it is more time consuming to be cracked. Therefore, 256-bit key size of AES presents the maximum security level than other algorithms such as AES-128 or DES (Data Encryption Standard).

Fig. 1 The encryption process of a video file by using 256-AES key.

Table 1 The required time to crack an algorithm with respect to its key size

Key Size	*Time to Crack*
56-bit (DES)	399 seconds
128-bit (AES)	$1{,}02 \times 10^{18}$ years
192-bit (AES)	$1{,}872 \times 10^{37}$ years
256-bit (AES)	$3{,}31 \times 10^{56}$ years

For some highly sensitive and important information, it is not always the best way to have a unique person in control of the key, and consequently, the security of the information. This problem is addressed by the development and use of secret sharing schemes, which allow keys to be shared among a group of people, with a predefined number of them needing to input their share in order to have access to the key [30]. Therefore, we introduce additionally a form of secret sharing scheme, Shamir's Secret Sharing (SSS) scheme, which is an encryption algorithm only for intra frames and described analytically by Vijayalakshmi et al. for MPEG videos [20]. Specifically, SSS scheme is a cryptographic algorithm in which a secret is shared into *n* unique parts for equal number participants, so as to be - some or all of them - necessary to reconstruct the original secret [20]. In our Encryption Algorithm for HEVC which is presented below, the embedded SSS algorithm is marked in italics.

Thus, by using these two algorithms, we ensure the encryption of only I frames, because P and B frames are useless without knowing the corresponding I frames [21]. Moreover, researches have shown that the encryption of only I frames can save 30-50% of encryption/decryption time and the size of encrypted stream does not change [21]. In the case of missing the secret key, the user's decoder will play quite different images from the original video, because of the fact that most of the image pixel values would have been changed [21].

Our proposed algorithm is formed as follows:

/*Proposed Encryption Algorithm for HEVC */
begin
open I-frame HEVC video file
create output file
*/*Encryption Algorithm: For Intra frame */*
for Each and Every DCT block
{
Step 1: Initialize nac = number of non-zero ACs
Step 2:
if (nac < 10 and nac > 5)
{
perform (4,5) secret sharing with
DC, AC1, . . . , AC3 as input and
store the result in DC, AC1, AC2, AC3, ACnac
}
if (nac < 20)
{
perform (8,9) secret sharing with
DC, AC1, . . . , AC7 as input and
store the result in DC, AC1, AC2, . . . , AC7, ACnac
}
if (nac > 20)
{
perform (12,13) secret sharing with
DC, AC1, . . . , AC11 as input and
store the result in DC, AC1, . . . , AC11, ACnac
}
}
while (not end of I-frame HEVC file)
{
read n bytes from input I-frame HEVC file in buffer
for each byte in buffer
{
if (collected sign bits == 256)
{
/*apply AES-256 encryption algorithm */
Rijndael(state,cipher_key)
{
key_expansion(cipher_key,expanded_key)
add_round_key(state,expaned_key)
/* Nr: Number of rounds,
Nc: No. of columns of state matrix */
for(i=1;i<Nr;i++)
Round(state,expaned_key + Nc*i)
Final_round(state,expanded_key+Nc*Nr)
}
put resulting sign bits in original place
}
}
write n bytes from buffer to output file
}
close input and output file
end

Table 2 The sizes of each sequence of the Test Classes after encryption with DES, AES-128 and AES-256 algorithms for HEVC and H.264 standards

Class	*Sequence name*	*Frame Count*	*Frame rate (fps)*	*Duration (sec)*	*Original Size HEVC - (H.264) (MB)*	*Size after encryption with DES HEVC - (H.264) (MB)*	*Size after encryption with AES-128 HEVC - (H.264) (MB)*	*Size after encryption with AES-256 HEVC - (H.264) (MB)*
A	***Traffic***	150	30	5	8,75 (24,31)	9,45 (26,25)	9,60 (26,67)	9,60 (26,67)
A	***PeopleOnStreet***	150	30	5	8,75 (24,31)	9,45 (26,25)	9,60 (26,67)	9,60 (26,67)
A	***Nebuta***	300	60	5	8,75 (24,31)	10,50 (29,17)	9,60 (26,67)	9,60 (26,67)
A	***SteamLocomotive***	300	60	5	8,75 (24,31)	10,50 (29.17)	9,60 (26,67)	9,60 (26,67)
B1	***Kimono***	240	24	10	7,50 (19,74)	8,40 (22,11)	7,68 (20,21)	7,68 (20,21)
B1	***ParkScene***	240	24	10	7,50 (19,74)	8,40 (22,11)	7,68 (20,21)	7,68 (20,21)
B2	***Cactus***	500	50	10	12,50 (32,89)	14,00 (36,84)	16,00 (42,10)	16,00 (42,10)
B2	***BQTerrace***	600	60	10	12,50 (32,89)	12,60 (33,15)	19,20 (50,52)	19,20 (50,52)
B2	***BasketballDrive***	500	50	10	12,50 (32,89)	14,00 (36,84)	16,00 (42,10)	16,00 (42,10)
C	***RaceHorses***	300	30	10	2,50 (5,68)	4,20 (9,54)	4,80 (10,91)	9,60 (21,81)
C	***BQMall***	600	60	10	2,50 (5,68)	4,20 (9,54)	9,60 (21,81)	19,20 (43,62)
C	***PartyScene***	500	50	10	2,50 (5,68)	3,50 (7,95)	8,00 (18,18)	16,00 (36,35)
C	***BasketballDrill***	500	50	10	2,50 (5,68)	3,50 (7,95)	8,00 (18,18)	16,00 (36,35)
D	***RaceHorses***	300	30	10	1,88 (3,91)	2,10 (4,38)	4,80 (10,01)	9,60 (20,02)
D	***BQSquare***	600	60	10	1,88 (3,91)	4,20 (8,76)	9,60 (20,02)	19,20 (40,04)
D	***BlowingBubbles***	500	50	10	1,88 (3,91)	3,50 (7,30)	8,00 (16,68)	16,00 (33,37)
D	***BasketballPass***	500	50	10	1,88 (3,91)	3,50 (7,30)	8,00 (16,68)	16,00 (33,37)

4 Experiments

In this section, we conducted tests upon specific video sequences so as to indicate diagrammatically the effect of cryptographic algorithms on them. Specifically, we calculated the size and the encryption time of the used video sequences, after the application of the cryptographic algorithms: DES, AES-128 and AES-256 respectively. Then, we consider our recommendations for each case relative to the protection level they provide.

Table 2 indicates the test sequences, retrieved by JCT-VC main configuration common conditions [22], which we used to conduct the experiments. As shown in this Table and in the following Table 3, the sequences differ from each other in terms of frame count, frame rate (fps) and bit rate (Mbps), and as a result of them, they have different size (MB) too.

Sequences used in the experiments are classified into five classes based on their resolution (class A, B1, B2, C, D). Class A sequences correspond to ultra high definition (UHD) sequences with a resolution of 2560x1600. Class B1 and B2 sequences correspond to full high definition (HD) sequences with a resolution of 1920x1080. Class C and Class D sequences correspond to WVGA and WQVGA resolutions of 800x480 and 400x240 respectively.

For the experiments, Class A includes the Traffic, PeopleOnStreet, Nebuta and SteamLocomotive sequences; Class B1 includes the Kimono, ParkScene sequences; Class B2 includes the Cactus, BQTerrace and BasketballDrive sequences; Class C includes the RaceHorses, BQMall, PartyScene and BasketballDrill sequences; and Class D includes the RaceHorses, BQSquare, BlowingBubbles and BasketballPass sequences.

For each Class (A, B1, B2, C, and D) we selected the maximum bitrate levels for the classes [22], as they are summarized in Table 3. This Table indicates also the transmission rate in MB per second for the ease of the

Table 3 Test Classes and bit rates for HEVC

Class	*Bit Rate*	*MB/sec*
A	14 Mbps	1,75MB
B1	6 Mbps	0,75MB
B2	10 Mbps	1,25MB
C	2 Mbps	0,25MB
D	1,5 Mbps	0,1875MB

following calculations.

For the calculation of the size of each video sequence, before and after encryption, we used the following general equation [23]:

$$CipherText = PlainText + Block - (PlainText\ MOD\ Block) \quad (1)$$

Thus, the above equation is modified as follows:

- For DES-56 encryption (56/8=7 blocks):

$$CipherSequence = PlainSequence + 7 - (PlainSequence\ MOD\ 7) \quad (2)$$

- For AES-128 encryption (128/8=16 blocks):

$$CipherSequence = PlainSequence + 16 - (PlainSequence\ MOD\ 16) \quad (3)$$

- For AES-256 encryption (256/8=32 blocks):

$$CipherSequence = PlainSequence + 32 - (PlainSequence\ MOD\ 32) \quad (4)$$

In addition, except of the above calculations upon HEVC standard, we present and compare the size of each sequence - before and after encryption - upon H.264 standard too. Based on previous subjective video performance comparisons [26], Class A' sequences compressed in HEVC standard (4K UHD) present an average 64% bitrate reduction compared to H.264, Class B' (1080p) 62%, Class C' (720p) 56% and Class D' (480p) 52% respectively.

Finally, a comparative analysis of the encryption speed of AES-128, AES-256 and DES algorithms for HEVC and H.264 standards, takes part in the next Section and is based on the following Table 4, which indicates the amount of encrypted data (MB) every second after the application of DES, AES-128 and AES-256 algorithm, respectively [29].

Table 4 The encryption speed of every algorithm

Algorithm	*Encryption Speed (MB/sec)*
AES Rijndael (128-bit key)	61,01
AES Rijndael (256-bit key)	48,23
DES (56-bit key)	21,34

5 Experimental Results

The results of the calculations of the previous section are presented in the Table 2. The following Figures 2-6 depict diagrammatically the changes of the size of each sequence of each class after the encryption with DES, AES-128 and AES-256 algorithms in H.264 and HEVC compression standards respectively. Specifically: Figure 2 depicts these changes of the size of each sequence of Class A; Figure 3 depicts these changes of the size of each sequence of Class B1; Figure 4 depicts these changes of the size of each sequence of Class B2; Figure 5 depicts these changes of the size of each sequence of Class C; Figure 6 depicts these changes of the size of each sequence of Class D.

Based on these diagrams, we highly recommend AES-256 for Class A, B1 and B2, while for classes C and D, AES-128 security level could be more convenient if the security factor is not the priority, because of the fact that in these classes, AES-256 increase very much the size of the relative video sequences compared to AES-128. Table 2 indicates clearly the accurate sizes of each sequence before and after the application of DES, AES-128 and AES-256 algorithms for H.264 and HEVC format respectively.

Generally, the AES-128 algorithm presents sufficient security level in some cases, but it is fact that AES-256 maximize very much the security level. On the other hand, DES algorithm seems to be inconvenient and unsafe, because it is regarded easy to be cracked (Table 1), while in some sequences increase their size more than the other two algorithms, as it is shown diagramatically. Moreover, bandwidth savings thanks to HEVC are confirmed from the above diagrams, not only to the original video sequences, but also to their encrypted forms after the effect of the cryptographic mechanism of DES, AES-128 and AES-256 algorithms, compared to H.264 compression standard.

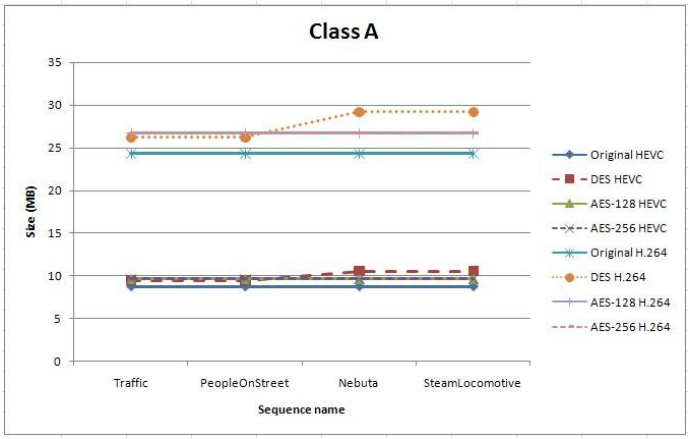

Fig. 1 The original size and the size after encryption with DES, AES-128 and AES-256 algorithms for the sequences of Class A.

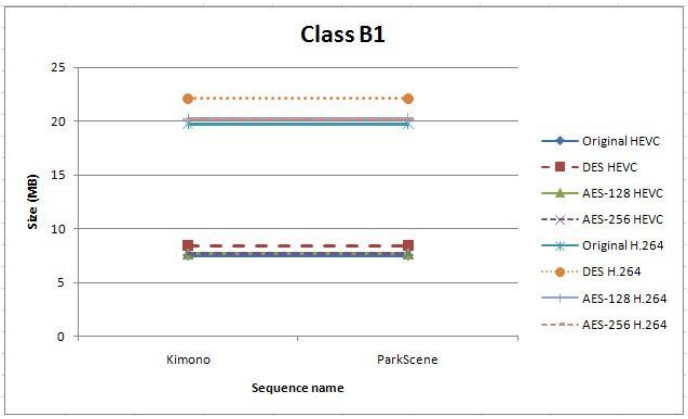

Fig. 2 The original size and the size after encryption with DES, AES-128 and AES-256 algorithms for the sequences of Class B1.

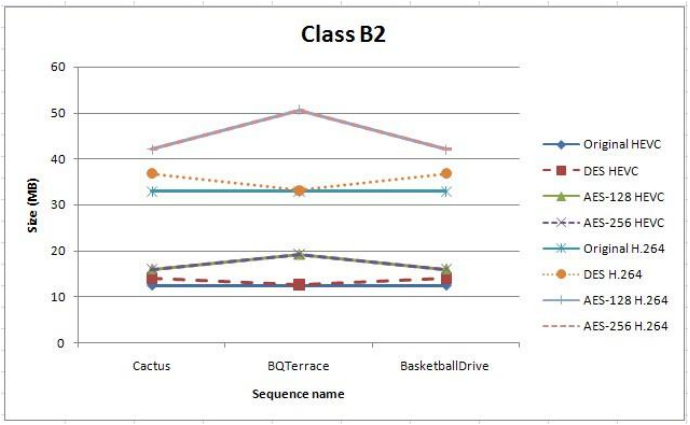

Fig. 3 The original size and the size after encryption with DES, AES-128 and AES-256 algorithms for the sequences of Class B2.

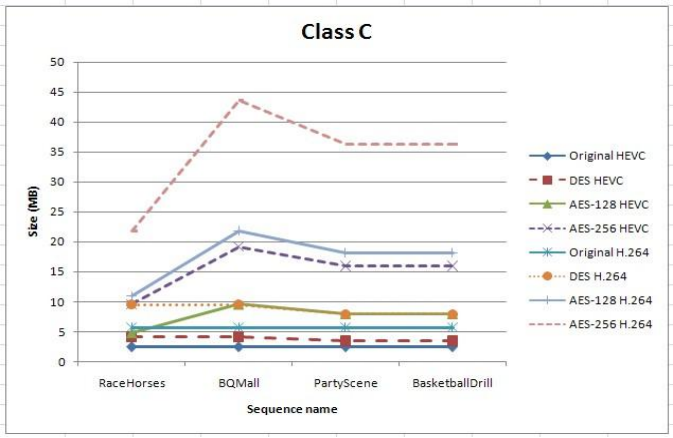

Fig. 4 The original size and the size after encryption with DES, AES-128 and AES-256 algorithms for the sequences of Class C.

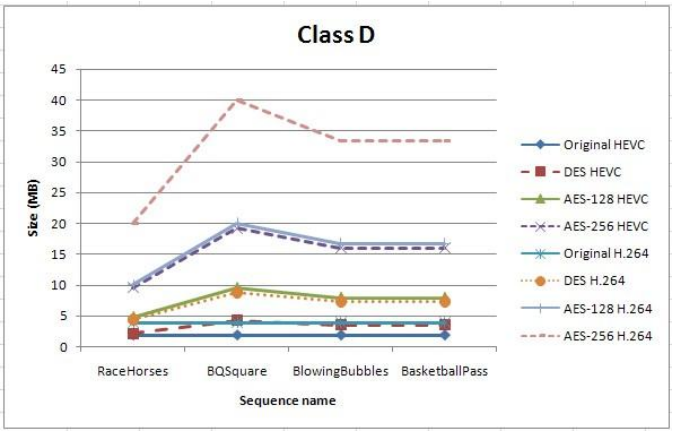

Fig. 5 The original size and the size after encryption with DES, AES-128 and AES-256 algorithms for the sequences of Class D.

In addition, the following Figures 7-11 depict diagrammatically the required time for the sequences of each class after the application of DES, AES-128 and AES-256 algorithms in H.264 and HEVC format respectively. Specifically, Figure 7 depicts these required encryption times for the sequences of Class A; Figure 8 depicts these required encryption times for the sequences of Class B1; Figure 9 depicts these required encryption times for the sequences of Class B2; Figure 10 depicts these required encryption times for the sequences of Class C; Figure 11 depicts these required encryption times for the sequences of Class D; for H.264 and HEVC compression standards respectively. The diagrams are the result of calculations based on Table 4.

According to these diagrams, AES-128 algorithm for HEVC standard is the faster, as it requires the minimum possible time to encrypt the video sequences of the Table 2. Specifically,

AES-128 needs 0.14, 0.12, 0.20, 0.04, and 0.03 seconds to encrypt each sequence of Class A, B1, B2, C, and D, respectively.

On the other hand, AES-256, despite the fact that it is slower than AES-128 about 20.95%, it offers much better security level, as it is depicted in Table 1. The corresponding encryption times for each sequence of Class A, B1, B2, C, and D are: 0.18, 0.16, 0.26, 0.05 and 0.04 seconds, respectively.

Moreover, DES algorithm is presented very slow as it can encrypt only 21.34 MB per second and thus, it is about 2,26 times slower than AES-256 and 2,86 times than AES-128 algorithm respectively.

Finally, the required encryption time for HEVC standard is much less than the corresponding time for H.264, due to the fact that HEVC presents bitrate reduction for all test classes compared to its previous compression standard, H.264 [26].

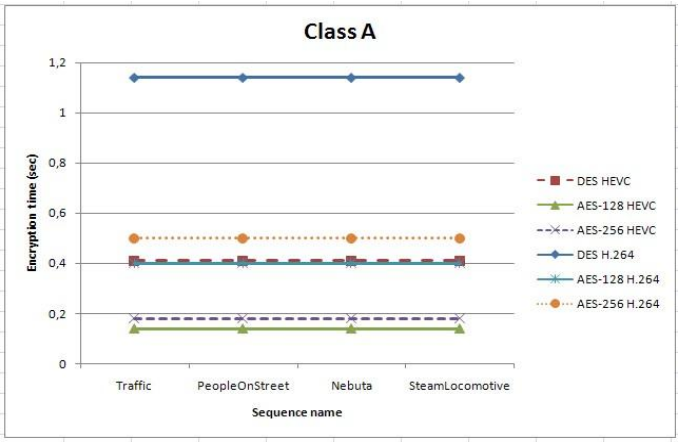

Fig. 7 The required encryption time of DES, AES-128 and AES-256 algorithms for the Class A sequences in HEVC and H.264 standard.

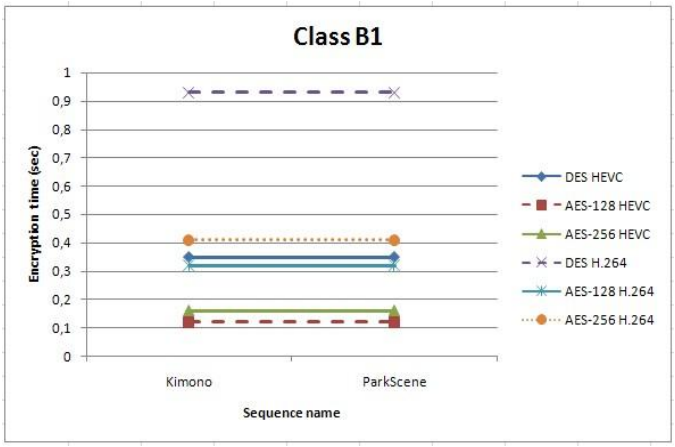

Fig. 8 The required encryption time of DES, AES-128 and AES-256 algorithms for the Class B1 sequences in HEVC and H.264 standard.

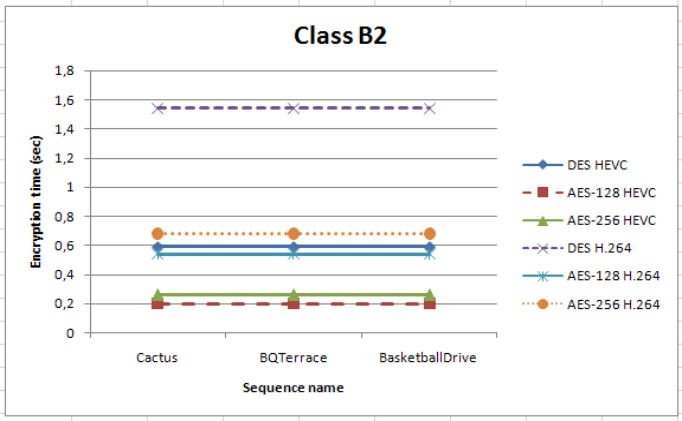

Fig. 9 The required encryption time of DES, AES-128 and AES-256 algorithms for the Class B2 sequences in HEVC and H.264 standard.

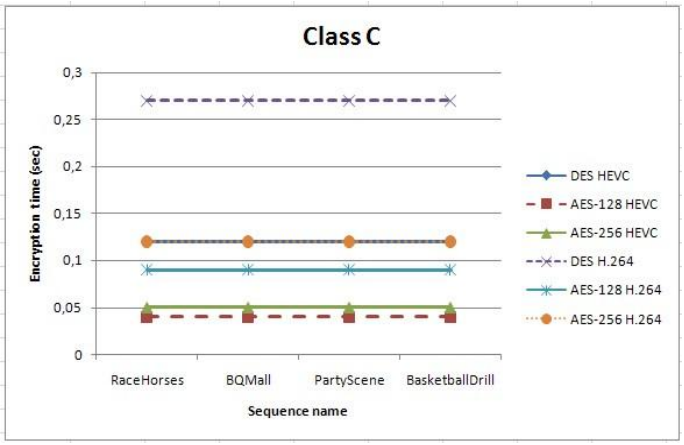

Fig. 10 The required encryption time of DES, AES-128 and AES-256 algorithms for the Class C sequences in HEVC and H.264 standard.

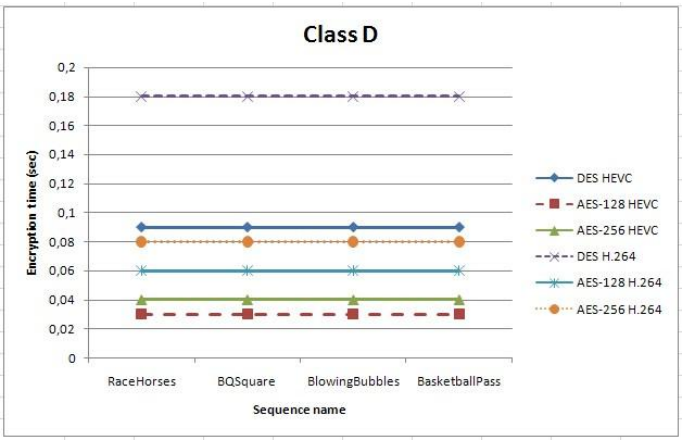

Fig. 11 The required encryption time of DES, AES-128 and AES-256 algorithms for the Class D sequences in HEVC and H.264 standard.

6 Conclusions and Future Work

A new encryption and transmission algorithm for efficient HEVC – media communications was presented. This algorithm merges two algorithms proposed for previous standards and it is modified so as to be amendable to the new video compression standard. A comparative analysis between DES, AES-128 and AES-256 was conducted to show which algorithm could be more convenient for the video sequences of each class A, B, C and D compressed with HEVC. Experimental results shows that despite the fact that AES-256 is slower than AES-128, it offers much better security level and it is better for the classes A and B, due to bandwidth factor, while AES-128 seems to be sufficient to encrypt video of the classes C and D if the security factor is not the priority.

In addition, according to the comparative analysis of the last two recent compression standards, HEVC compression standard is shown better than H.264 using the same algorithms - DES, AES-128 and AES-256 - for intra frames encryption, because of the fact that it presents bandwidth savings and requires less time to be encrypted with relevant algorithms compared to H.264. Future work will include comparative analysis of the effectiveness of our algorithm with the other proposed algorithms for HEVC compression standard.

References

[1] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han and Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No.12, December 2012.

[2] Dan Grois, Detlev Marpe, Amit Mulayoff, Benaya Itzhaky and Ofer Hadar, "Performance Comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC Encoders", 30th Picture Coding Symposium (PCS), December 2013.

[3] Jens-Rainer Ohm, Gary J. Sullivan, Heiko Schwarz, Thiow Keng Tan and Thomas Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No.12, December 2012.

[4] Jarno Vanne, Marko Viitanen, Timo D. Hamalainen and Antti Hallapuro, "Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No.12, December 2012.

[5] Frank Bossen, Benjamin Bross, Karsten Suhring and David Flynn, "HEVC Complexity and Implementation Analysis", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No.12, December 2012.

[6] Thomas Schierl, Miska M. Hannuksela, Ye-Kui Wang and Stephan Wenger, "System Layer Integration of High Efficiency Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No.12, December 2012.

[7] Philippe Hanhart, Martin Rerabek, Francesca De Simone, and Touradj Ebrahimi , "Subjective quality evaluation of the upcoming HEVC video compression standard", Applications of Digital Image Processing XXXV, Proceedings of SPIE, Vol. 8499 84990V, October 2012.

[8] James Nightingale, Qi Wang and Christos Grecos, "Benchmarking Real-Time HEVC Streaming", Real-Time Image and Video Processing, Proceedings of SPIE, Vol. 8437 84370D-1, 2012.

[9] Shanshe Wang, Siwei Ma, Shiqi Wang, Debin Zhao and Wen Gao, "Rate-GOP Based Rate Control for High Efficiency Video Coding", IEEE Journal of Selected Topics in Signal Processing, Vol. 7, No.6, December 2013.

[10] Fuwen Liu and Hartmut Koenig, "A survey of video encryption algorithms", Computers & Security, Vol. 29, Issue 1, pp. 3-15, February 2010.

[11] Cyril Bergeron and Catherine Lamy-Bergot, "Compliant selective encryption for H.264/AVC video streams", IEEE 7th Workshop on Multimedia Signal Processing, December 2005.

[12] Saranya P. and Varalakshmi L.M.,"H.264 based Selective Video Encryption for Mobile Applications", International Journal of Computer Applications, Vo. 17, No. 4, March 2011.

[13] Thomas Stütz and Andreas Uhl, "A Survey of H.264 AVC/SVC Encryption", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, Issue: 3, pp. 325 - 339, March 2012.

[14] Guo Jie, Qiu Weidong, Du Chao and Chen Kefei, "A Scalable Video Encryption Algorithm for H.264/SVC", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE), 2013.

[15] Loic Dubois, Zafar Shahid and William Puech, "Selective Encryption of Images and Videos: From JPEG to H.265/HEVC through JPEG2000 and H.264/AVC", Progress in Data Encryption Research, 137-178, 2013.

[16] Heinz Hofbauer, Andreas Uhl and Andreas Unterweger, "Transparent Encryption for HEVC Using Bit-Stream-based Selective Coefficient Sign Encryption", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1986-1990, May 2014.

[17] Zafar Shahid and William Puech, "Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings", IEEE Transactions on Multimedia, January 2013.

[18] Glenn Van Wallendael, Andras Boho, Jan De Cock, Adrian Munteanu and Rik Van de Walle, "Encryption for High Efficiency Video Coding with Video Adaptation Capabilities", IEEE International Conference on Consumer Electronics (ICCE), pp. 31-32, January 2013.

[19] Zafar Shahid and William Puech, "Investigating the Structure Preserving Encryption of High Efficiency Video Coding (HEVC)", Proceedings SPIE, Vol. 8656, Real-Time Image and Video Processing, June 2013.

[20] V. Vijayalakshmi, L.M. Varalakshmi, and G.F. Sudha, "Efficient Encryption of Intra and Inter Frames in MPEG Video", Recent Trends in Network Security and Applications Communications in Computer and Information Science, Vol. 89, pp 93-104, July 2010.

[21] Jayshri Nehete, K. Bhagyalakshmi, M.B. Manjunath, Shashikant Chaudhari and T.R. Ramamohan, "A Real-time MPEG Video Encryption Algorithm using AES", The National Conference on Communications (NCC), pp. 164-168, 2003.

[22] F. Bossen, "Common HM test conditions and software reference configurations," document JCTVC-L1100 of JCT-VC, Geneva, CH, Jan. 2013.

[23] "How to Calculate the Size of Encrypted Data?", http://www.obviex.com/articles/CiphertextSize.pdf

[24] Westlund and Harold B., "NIST reports measurable success of Advanced Encryption Standard, Journal of Research of the National Institute of Standards and Technology, 2002.

[25] Mohit Arora, "How secure is AES against brute force attacks?", http://www.eetimes.com/document.asp?doc_id=1279619

[26] TK Tan, Marta Mrak, Vittorio Baroncini and Naeem Ramzan, "Report on HEVC compression performance verification testing", Joint Collaborative Team on Video Coding (JCT-VC), 2014.

[27] Mahsa T. Pourazad, Colin Doutre, Maryam Azimi and Panos Nasiopoulos, "HEVC: The New Gold Standard for Video Compression, How does HEVC compare with H.264/AVC?", IEEE Consumer Electronics Magazine, July 2012.

[28] Alex Lee, DongSan Jun, Jongho Kim, Jin Soo Choi, and Jinwoong Kim, "An Efficient Inter Prediction Mode Decision Method for Fast Motion Estimation in High Efficiency Video Coding", IEEE International Conference on ICT Convergence (ICTC), pp. 502-505, October 2013.

[29] Abdel-Karim Al Tamimi, "Performance Analysis of Data Encryption Algorithms", http://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/

[30] Russ Martin, "Introduction to Secret Sharing Schemes", Computer Science Department, Rochester Institute of Technology, May 2012.