

Video Steganography Encoding H.264/AVC Video Streams By Codeword Substitution

Shirin Khan

MTECH Student

Dept. of Electronics and communication engg

Anjuman College of Engg and Tech

Sadar, Nagpur

Shirin.khan210@gmail.com

Prof. M. T. Hasan

Assistant Professor

Dept. of Electronics and communication engg

Anjuman College of Engg and Tech

Sadar, Nagpur

mthasan21@gmail.com

Abstract— With the rapid development of various multimedia technologies, more and more multimedia data are generated and transmitted in the medical, commercial, and military fields, which may include some sensitive information which should not be accessed by or can only be partially exposed to the general users. Therefore, security and privacy has become an important issue. Video steganography techniques can be used to embed a secret message and secret image into a video bit stream for copyright protection, access control and transaction tracking. Digital video sometimes needs to be stored and processed in an encrypted format to maintain security and privacy. For the purpose of content notation and/or tampering detection, it is necessary to perform data hiding in these encrypted videos. In this way, data hiding in encrypted domain without decryption preserves the confidentiality of the content. In addition, it is more efficient without decryption followed by data hiding and re-encryption. A novel scheme of data hiding directly in the encrypted version of H.264/AVC video stream is proposed. MPEG-4 Part-10 is nothing but H.264/AVC (Advanced video coding). It includes the following three parts, i.e., H.264/AVC video encryption, data embedding and data extraction. By analyzing the property of H.264/AVC codec, the codewords of intraprediction modes, the codewords of motion vector differences, and the codewords of residual coefficients are encrypted with stream ciphers. Then, a data hider may embed additional data in the encrypted domain by using codeword substitution technique, without knowing the original video content. In order to adapt to different application scenarios, data extraction can be done either in the encrypted domain or in the decrypted domain. Furthermore, video file size is strictly preserved even after encryption and data embedding.

Keywords: *Data hiding, Video Streaming, H.264/AVC, Codeword substitutions.*

I. INTRODUCTION

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital Videos are the most popular because of their frequency on the Internet. For hiding secret information in Videos, there exist a large variety of steganography techniques some are more complex than others and all of them have respective strong and weak points. The capability of performing data hiding directly in encrypted H.264/AVC video streams would avoid the leakage of video content, which can help address the security and privacy concerns with cloud computing. Cloud computing has become an important technology trend, which can provide highly efficient computation and large-scale storage solution for video data. Given that cloud services may attract more attacks and are vulnerable to untrustworthy system administrators, it is desired that the video content is accessible in encrypted form. For example, a cloud server can embed the additional information (e.g., video notation, or authentication data) into an encrypted version of an H.264/AVC video by using data hiding technique. With the hidden information, the server can manage the video or verify its integrity without knowing the original content, and thus the security and privacy can be protected.

In addition to cloud computing, this technology can also be applied to other prominent application scenarios. For example, when medical videos or surveillance videos have been encrypted for protecting the privacy of the people, a database manager may embed the personal information into the corresponding encrypted videos to provide the data management capabilities in the encrypted domain. In

watermarking scheme in the encrypted domain using Paillier cryptosystem is proposed based on the security requirements of buyer-seller watermarking protocols. However, due to the constraints of the Paillier cryptosystem, the encryption of an original image results in a high overhead in storage and computation. The encryption is performed by using bit-XOR (exclusive-OR) operation. In these methods, however, the host image is in an uncompressed format. As said the above mentioned works have been focused on image. With the increasing demands of providing video data security and privacy protection, data hiding in encrypted H.264/AVC videos will undoubtedly become popular in the near future. Obviously, due to the constraint of the underlying encryption, it is very difficult and sometimes impossible to transplant the existing data hiding algorithms to the encrypted domain.

For example, during H.264/AVC compression, the intra-prediction mode (IPM), motion vector difference (MVD) and DCT coefficients' signs are encrypted, while DCT coefficients' amplitudes are watermarked adaptively. A combined scheme of encryption and watermarking is presented, which can provide the access right as well as the authentication of video content simultaneously. In summary, in the existing related technologies, encryption and data embedding are implemented almost simultaneously during H.264/AVC compression process. However, to meet the aforementioned application requirements, it's necessary to perform data hiding directly in the encrypted domain. In addition, the earlier approaches do not operate on the compressed bitstream. That is, encryption and watermark embedding are accomplished in the encoding process, while decryption and watermark detection are completed in the decoding process. The compression/decompression cycle is

time-consuming and hampers real-time implementation. Besides, encryption and watermark embedding would lead to increasing the bit-rate of H.264/AVC bitstream. Therefore, it becomes highly desirable to develop data hiding algorithms that work entirely on encoded bitstream in the encrypted domain. However, there are some significant challenges for data hiding directly in compressed and encrypted bitstream. The first challenge is to determine where and how the bitstream can be modified so that the encrypted bitstream with hidden data is still a compliant compressed bitstream. The second challenge is to insure that decrypted videos containing hidden data can still appear to be of high visual fidelity. The third challenge is to maintain the file size after encryption and data hiding, which requires that the impact on compression gain is minimal. The fourth challenge is that the hidden data can be extracted either from the encrypted video stream or from the decrypted video stream, which is much more applicable in practical applications.

Based on the analysis given above, we propose a novel scheme to embed secret data directly in compressed and then encrypted H.264/AVC bitstream. Firstly, by analyzing the property of H.264/AVC codec, the codewords of IPMs, the codewords of MVDs, and the codewords of residual coefficients are encrypted with a stream cipher. The encryption algorithm is combined with the Exp-Golomb entropy coding and Context-adaptive variable-length coding (CAVLC), which keeps the codeword length unchanged. Then, data hiding in the encrypted domain is performed based on a novel codeword substituting scheme.

In contrast to the existing technologies, the proposed scheme can achieve excellent performance in the following three different prospects.

- The data hiding is performed directly in encrypted H.264/AVC video bitstream.
- The scheme can ensure both the format compliance and the strict file size preservation.
- The scheme can be applied to two different application scenarios by extracting the hidden data either from the encrypted video stream or from the decrypted video stream.

II. OBJECTIVE

- The main objective of this project is to maintain security and privacy.
- The hidden data can be extracted either from the encrypted video stream or from the decrypted video stream, which is much more applicable in practical applications.
- The main objective is to enhance compression performance and provides a provision of a network friendly video representation addressing conversational applications.

H.264/AVC has achieved a significant improvements in rate distortion efficiency relative to existing standards H.264/AVC covers all common video conferencing and high definition video storage. To address the need for flexibility and customizability, the H.264/AVC design covers a video coding layer (VCL), which is designed to efficiently represent the video content, and a network subtraction layer (NAL) which formats the VCL representation in a manner appropriate for conveyance by a variety of transport layer or storage

media. Relative to prior video coding methods, as exemplified by MPEG-2 video, some highlighted features of the design that enable enhanced coding efficiency include the following enhancement of the ability to predict the values of the content of a pictures to be encoded.

1. Variable block size motion compensation with small block sizes.
2. Quarter sample accurate motion compensation.
3. Motion vectors over picture boundaries.
4. Multiple reference picture motion compensation.
5. Decoupling of reference order from display order.
6. Decoupling of picture representation methods from picture referencing capability.
7. Weighted prediction.
8. Improved skipped and direct motion inference.
9. Directional spatial prediction for intra coding.
10. In the loop deblocking filtering.

III. PROPOSED SCHEME

In this section, a novel scheme of data hiding in the encrypted version of H.264/AVC videos is presented, which includes three parts, i.e., H.264/AVC video encryption, data embedding and data extraction. The content owner encrypts the original H.264/AVC video stream using standard stream ciphers with encryption keys to produce an encrypted video stream. Then, the data-hider (e.g., a cloud server) can embed the additional data into the encrypted video stream by using codeword substituting method, without knowing the original video content. At the receiver end, the hidden data extraction can be accomplished either in encrypted or in decrypted version. The diagram of the proposed framework is shown in Fig. 1, where the encryption and data embedding are depicted in part (a), and the data extraction and video decryption are shown in part (b).

A. Encryption of H.264/AVC Video Stream

Video encryption often requires that the scheme be time efficient to meet the requirement of real time and format compliance. It is not practical to encrypt the whole compressed video bitstream like what the traditional ciphers do because of the following two constraints, i.e., format compliance and computational cost. Alternatively, only a fraction of video data is encrypted to improve the efficiency while still achieving adequate security. The key issue is then how to select the sensitive data to encrypt. It is reasonable to encrypt both spatial information (IPM and residual data) and motion information (MVD) during H.264/AVC encoding.

In this paper, an H.264/AVC video encryption scheme with good performance including security, efficiency, and format compliance is proposed. By analyzing the property of H.264/AVC codec, three sensitive parts (i.e., IPMs, MVDs, and residual coefficients) are encrypted with stream ciphers. Compared with present work, the proposed encryption algorithm is performed not during H.264/AVC encoding but in the H.264/AVC compressed domain. In this case, the bit-stream will be modified directly. Selective encryption in the H.264/AVC compressed domain has been already presented on context-adaptive variable length coding (CAVLC) and context-

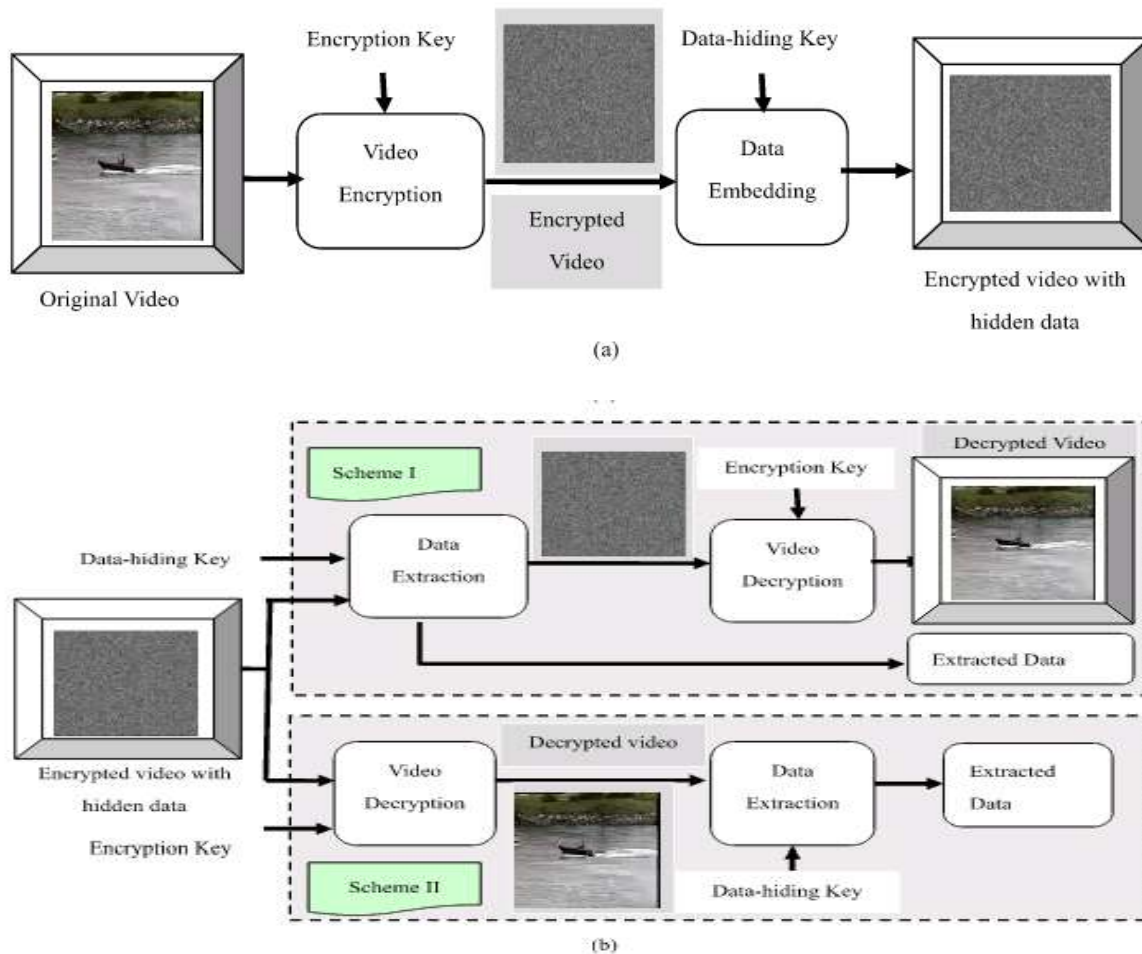


Fig.1 Diagram of proposed scheme. (a) Video encryption and data embedding at the sender end. (b) Data extraction and video display at the receiver end in two scenarios.

adaptive binary arithmetic coding (CABAC). In this paper, we have improved and enhanced the previous proposed approach by encrypting more syntax elements. We encrypt the codewords of IPMs, the codewords of MVDs, and the codewords of residual coefficients. The encrypted bitstream is still H.264/AVC compliant and can be decoded by any standard-compliant H.264/AVC decoder, but the encrypted video data is treated completely different compared to plain text video data. In fact, performing the format-compliant encryption directly on the compressed bitstream is extremely complicated as the internal states of the encoder have to be preserved, otherwise the remaining data is interpreted falsely which may easily lead to format violations.

In H.264/AVC baseline profile, CAVLC entropy coding is used to encode the quantized coefficients of a residual block. Each CAVLC codeword can be expressed as the following format:

$\{Coeff_token, Sign_of_TrailingOnes, Level, Total_zeros, Run_before\}$

To keep the bitstream compliant, not all syntax elements can be modified during encryption process. For example, *Coeff_token*, *Total_zeros*, and *Run_before* should remain

unchanged. Therefore, residual data encryption can be accomplished by modifying the codewords of *Sign_of_TrailingOnes* and *Level*. The *Sign_of_TrailingOnes*

is encoded with a single bit. Bit “0” is assigned for +1 and bit “1” is assigned for -1. The codeword of *Sign_of_TrailingOnes* is encrypted by applying the bitwise XOR operation with a standard stream cipher, which is determined by an encryption key *E_Key4*.

The codeword for each *Level* is made up of a prefix (*level_prefix*) and a suffix (*level_suffix*) as

$$Level\ codeword = [level\ prefix], [level\ suffix]$$

Table 1 shows *Levels* with different *suffixLength* and corresponding codewords. The last bit of the codeword is encrypted by applying the bitwise XOR operation with a standard stream cipher, which is determined by an encryption key *E_Key5*. According to Table 1, the last bit encryption may change the sign of *Levels*, but does not affect the length of the codeword and satisfies the format compliance. For example, when *suffixLength* is equal to 1, the codewords corresponding to “2” and “-2” are “010” and “011”, respectively, which have the same length. It should be noted that when *suffixLength* is

equal to 0, the codewords should keep unchanged during the encryption process.

Table 1 Levels and Corresponding Codewords

S	Level (>0)	Codeword	Level(<0)	Codeword
0	1	1	-1	01
	2	001	-2	0001
	3	00001	-3	000001
	4	0000001	-4	00000001
1	1	10	-1	11
	2	010	-2	011
	3	0010	-3	0011
	4	00010	-4	00011
	5	000010	-5	000011
	6	0000010	-6	0000011
	7	00000010	-7	00000011
	8	000000010	-8	000000011
2	1	100	-1	101
	2	110	-2	111
	3	0100	-3	0101
	4	0110	-4	0111
	5	00100	-5	00101
	6	00110	-6	00111
	7	000100	-7	000101
	8	000110	-8	000111
	9	0000100	-9	0000101
	10	0000110	-10	0000111
	11	00000100	-11	00000101
	12	00000110	-12	00000111
	13	000000100	-13	000000101
	14	000000110	-14	000000111
3	1	1000	-1	1001
	2	1010	-2	1011
	3	1100	-3	1101
	4	1110	-4	1111
	5	01000	-5	01001
	6	01010	-6	01011
	7	01100	-7	01101
	8	01110	-8	01111
	9	001000	-9	001001
	10	001010	-10	001011
	11	001100	-11	001101
	12	001110	-12	001111
	13	0001000	-13	0001001
	14	0001010	-14	0001011

B. Data Embedding

In the encrypted bitstream of H.264/AVC, the proposed data embedding is accomplished by substituting eligible codewords of Levels in Table 1. Since the sign of Levels are encrypted, data hiding should not affect the sign of Levels. Besides, the codewords substitution should satisfy the following three limitations. First, the bitstream after codeword substituting must remain syntax compliance so that it can be decoded by standard decoder. Second, to keep the bit-rate

unchanged, the substituted codeword should have the same size as the original codeword. Third, data hiding does cause visual degradation but the impact should be kept to minimum. That is, the embedded data after video decryption has to be invisible to a human observer. So the value of Level corresponding to the substituted codeword should keep close to the value of Level corresponding to the original codeword. In addition, the codewords of Levels within P-frames are used for data hiding, while the codewords of Levels in I-frames are remained unchanged. Because I-frame is the first frame in a group of pictures (GOPs), the error occurred in I-frame will be propagated to subsequent P-frames.

According to the analysis given above, we can see that there are no corresponding substituted codewords when suffixLength is equal to 0 or 1, as shown in Table 1. When suffixLength is equal to 0, we cannot find a pair of codewords with the same size. When suffixLength is equal to 1, one codeword also cannot be substituted by another codeword with the same size, since this substitution would change the sign of Level. Then the codewords of Levels which suffixLength is 2 or 3 would be divided into two opposite codespaces denoted as C0 and C1 as shown in Fig.2.

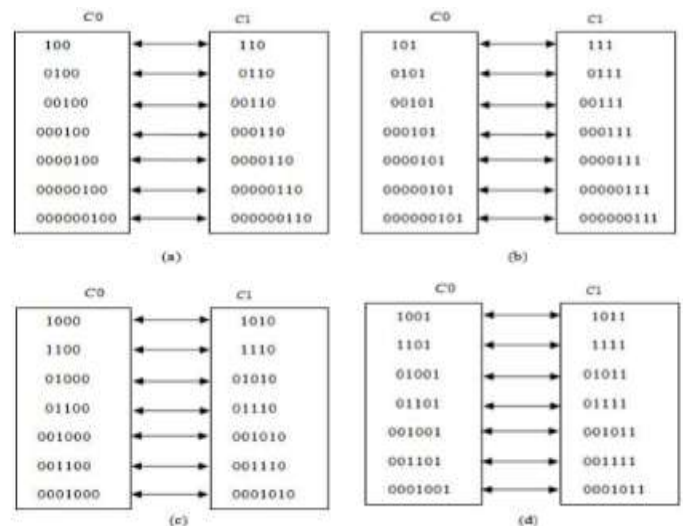


Fig. 2 CAVLC codeword mapping (a) Suffix Length = 2& Level > 0. (b) Suffix Length = 2& Level > 0. (c) Suffix Length =3& Level > 0. (d) Suffix Length =3& Level < 0.

The codewords assigned in C0 and C1 are associated with binary hidden information “0” and “1”. Suppose the additional data that we want to embed is a binary sequence denoted as $B = \{ b(i) | i = 1, 2, \dots, L, b(i) \in \{0, 1\} \}$. Data hiding is performed directly in encrypted bitstream through the following steps.

Step1. In order to enhance the security, the additional data is encrypted with the chaotic pseudorandom sequence $P = \{ p(i) | i = 1, 2, \dots, L, p(i) \in \{0, 1\} \}$ to generate the to-be-embedded sequence $W = \{ w(i) | i = 1, 2, \dots, L, w(i) \in \{0, 1\} \}$. The sequence P is generated by using logistic map with an initial value that is the data hiding key. It is very difficult for anyone who does not retain the data hiding key to recover the additional data.

Step2. The codewords of Levels are obtained by parsing the encrypted H.264/AVC bitstream.

Step3. If current codeword belongs to codespaces C0 or C1, the to-be-embedded data bit can be embedded by codeword substituting. Otherwise, the codeword is left unchanged. The

detailed procedure of codeword substituting for data hiding is shown in Fig. 3. For example, when Level is positive 1 and its suffixLength is 3, then its corresponding codeword is “1000” which belongs to C0 as shown in Fig. 2(c). If the data bit “1” needs to be embedded, the codeword “1000” should be replaced with “1010”. Otherwise, if the data bit “0” needs to be embedded, the codeword “1000” will keep unchanged.

Step4. Choose the next codeword and then go to Step3 for data hiding. If there are no more data bits to be embedded, the embedding process is stopped.

Suppose the to-be-embedded data is “1001”, the CAVLC codeword of Level parsing from H.264/AVC bitstream is “01 010 00100 00100 0001011 0000100” and the encryption stream is “10111”, an example of data embedding based on codeword mapping is shown in Fig. 4(a).

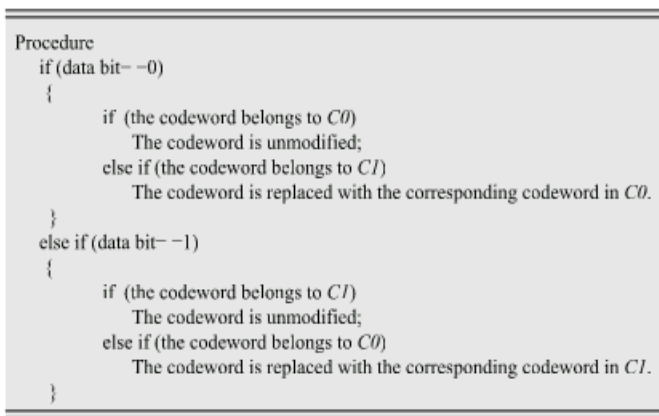


Fig. 3 The procedure of codeword mapping.

C. Data Extraction

In this scheme, the hidden data can be extracted either in encrypted or decrypted domain, as shown in Fig.1(b). Data extraction process is fast and simple. We will first discuss the extraction in encrypted domain followed by decrypted domain.

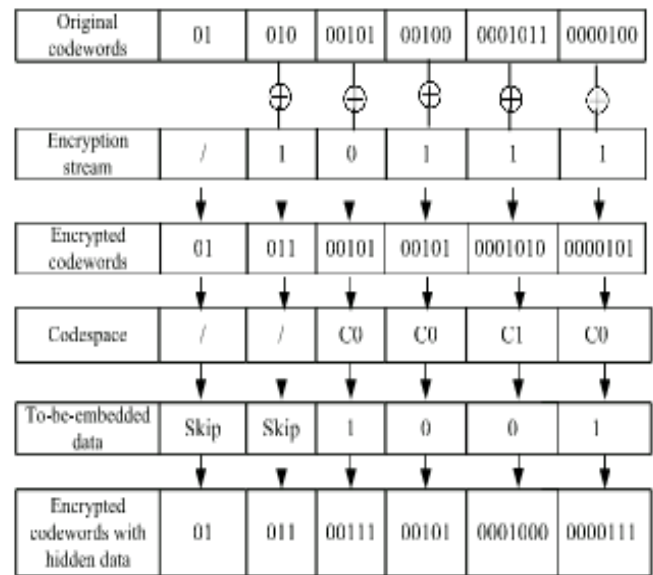
1) Scheme 1: Encrypted Domain Extraction.

To protect privacy, a database manager (e.g., cloud server) may only get access to the data hiding key and have to manipulate data in encrypted domain. Data extraction in encrypted domain guarantees the feasibility of our scheme in this case. In encrypted domain, as shown in Fig.1(b), encrypted video with hidden data is directly sent to the data extraction module, and the extraction process is given as follows.

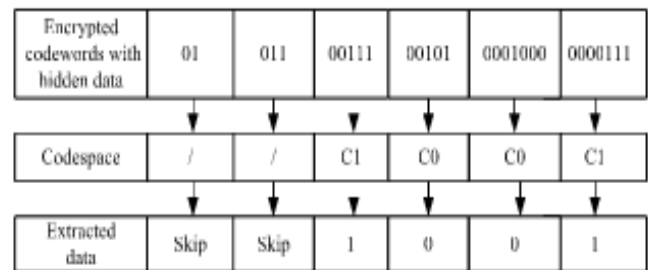
Step1: The codewords of Levels are firstly identified by parsing the encrypted bitstream.

Step2: If the codeword belongs to codespace C0, the extracted data bit is “0”. If the codeword belongs to codespace C1, the extracted data bit is “1”.

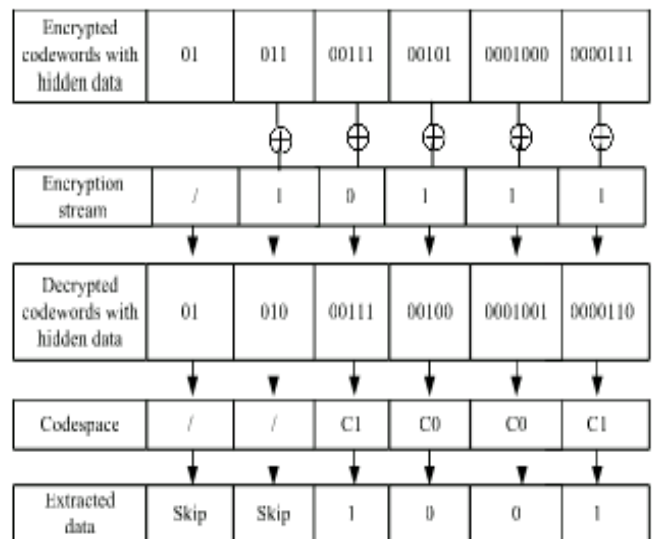
Step3: According to the data hiding key, the same chaotic pseudo-random sequence P that was used in the embedding process can be generated. Then the extracted bit sequence could be decrypted by using P to get the original additional information. Since the whole process is entirely operated in encrypted domain, it effectively avoids the leakage of original video content. An example of data extraction in encrypted domain is shown in Fig. 4(b).



(a)



(b)



(c)

Fig.4 An example of data embedding and extraction (a) Data embedding. (b) Data extraction in encrypted domain. (c) Data extraction in decrypted domain.

2) *Scheme II: Decrypted Domain Extraction.*

In scheme I, both embedding and extraction of the data are performed in encrypted domain. However, in some cases, users want to decrypt the video first and extract the hidden data from the decrypted video. For example, an authorized user, which owned the encryption key, received the encrypted video with hidden data. The received video can be decrypted using the encryption key. That is, the decrypted video still includes the hidden data, which can be used to trace the source of the data. Data extraction in decrypted domain is suitable for this case. As shown in Fig. 1(b), the received encrypted video with hidden data is first pass through the decryption module. The whole process of decryption and data extraction is given as follows.

Step1: Generate encryption streams with the encryption keys as given in encryption process.

Step2: The codewords of IPMs, MVDs, Sign_of_TrailingOnes and Levels are identified by parsing the encrypted bitstream.

Step3: The decryption process is identical to the encryption process, since XOR operation is symmetric. The encrypted codewords can be decrypted by performing XOR operation with generated encryption streams, and then two XOR operations cancel each other out, which renders the original plain-text. Since the encryption streams depend on the encryption keys, the decryption is possible only for the authorized users. After generating the decrypted codewords with hidden data, the content owner can further extract the hidden information.

Step4: According to Table 1, the last bit encryption may change the sign of Level. However, as shown in Fig.2, the encrypted codeword and the original codeword are still in the same codespaces. If the decrypted codeword of Level belongs to codespace C0, the extracted data bit is “0”. If the decrypted codeword of Level belongs to codespace C1, the extracted data bit is “1”.

Step5: Generate the same pseudorandom sequence P that was used in embedding process according to the data hiding key. The extracted bit sequence should be decrypted to get the original additional information. An example of data extraction in decrypted domain is shown in Fig.4(c).

IV. EXPERIMENT RESULT

The proposed data hiding scheme has been implemented in the H.264/AVC standard coding technique with MATLAB reference software. Three well-known standard video sequences (i.e., Movie, Mobile, Hall, and News) in QCIF format (176×144) at the frame rate 30 frames/s are used for simulation. We can use all frame of a video file but the first 50 frames in each video sequence are used in the experiments. Here, we successfully secured our data in this video steganography form.

Besides subjective observation, PSNR (Peak Signal to Noise Ratio), RMSE (Root Mean Squared Error), CR (Compression Ratio), BPP (Bits per Second) and Accuracy have been adopted to evaluate the perceptual quality. All these values in each video are shown in Table 2. PSNR is widely used objective video quality metric. However, it does not perfectly correlate with a perceived visual quality due to nonlinear behavior of human visual system.

Table 2 Some Parameters Based On Resultant Videos

VIDEOS	PSNR	ACCURACY	RMSE	CR	BPP
V1	19.5153	96.2709	3.7291	1.4156e+04	9.5367e-04
	19.5162	96.2724	3.7276	1.4156e+04	9.5367e-04
	19.5497	96.2708	3.7292	1.4156e+04	9.5367e-04
	19.5439	96.2713	3.7287	1.4156e+04	9.5367e-04
	19.5491	93.2705	3.7295	1.4156e+04	9.5367e-04
V2	28.3562	96.0551	3.9449	1.4156e+04	9.5367e-04
	28.3562	96.0551	3.9449	1.4156e+04	9.5367e-04
	28.3562	96.0551	3.9449	1.4156e+04	9.5367e-04
	28.3562	96.0551	3.9449	1.4156e+04	9.5367e-04
	28.3405	96.0542	3.9458	1.4156e+04	9.5367e-04
V3	27.6492	96.0507	3.9493	1.4156e+04	9.5367e-04
	27.6492	96.0507	3.9493	1.4156e+04	9.5367e-04
	27.6492	96.0507	3.9493	1.4156e+04	9.5367e-04
	27.6492	96.0507	3.9493	1.4156e+04	9.5367e-04
	27.5588	96.0499	3.9501	1.4156e+04	9.5367e-04



Fig. 5 Original video frames.



Fig. 7 Encrypted video frames with hidden data.



Fig. 6 Encrypted video frames.



Fig. 8 Decrypted video frames with hidden data.



V. DISCUSSION

As described in Table 2, for low motion sequence (such as Hall, News), the embedding capacity is low if only the codewords of Levels are used for data hiding. In this case, both the CAVLC codewords of Levels and the Exp-Golomb codewords of MVDs can be used for data hiding. For this type of video, the degradation in video quality caused by data hiding is quite small. So the combination is entirely feasible. The embedding capacity is improved obviously, but the video quality degradation is also negligible. Based on the above analysis, we can make a flexible choice of embedding carrier according to the situation. To the best of our knowledge, till now, there is no algorithm to embed additional data directly in encrypted H.264/AVC video stream. Therefore, no detailed experimental comparisons are given. As described in previous section, in the existing related technologies, encryption and

data hiding are accomplished almost simultaneously during H.264/AVC encoding process. In addition, encryption and data embedding would lead to increasing the bit-rate of H.264/AVC bitstream. On the contrary, our proposed scheme can encrypt H.264/AVC video stream directly and then embeds data into encrypted H.264/AVC video stream to meet the privacy-preserving requirements. The bit-rate of the encrypted H.264/AVC video stream containing hidden data is exactly the same as the original H.264/AVC video stream.

VI. CONCLUSION

Data hiding is a new technique that had drawn attention because of the privacy-preserving. Steganography and visual cryptographic is used for data hiding and extraction of data. An algorithm to embed additional data in encrypted H.264/AVC bitstream is presented, which consists of video encryption, data

embedding and data extraction phases. The algorithm can preserve the bit-rate exactly even after encryption and data embedding, and is simple to implement as it is directly performed in the compressed and encrypted domain, i.e., it does not require decrypting or partial decompression of the video stream thus making it ideal for real-time video applications. The data-hider can embed additional data into the encrypted bitstream using codeword substituting, even though he does not know the original video content. Since data hiding is completed entirely in the encrypted domain, our method can preserve the confidentiality of the content completely. With an encrypted video containing hidden data, data extraction can be carried out either in encrypted or decrypted domain, which provides two different practical applications. Another advantage is that it is fully compliant with the H.264/AVC syntax. Experimental results have shown that the proposed encryption and data embedding scheme can preserve file-size, whereas the degradation in video quality caused by data hiding is quite small.

REFERENCES

- [1] Dawen Xu, Rangding Wang, & Yun Q Shi, "Data hiding in encrypted H.264/AVC video streams by codeword substitution", *IEEE Trans. Inf. Forensics Security*, vol.9, No.4, pp.596-606, Apr.2014
- [2] W. J. Lu, A. Varna, and M. Wu, "Secure video processing: Problems and challenges," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Prague, Czech Republic, May 2011, pp. 5856–5859.
- [3] B. Zhao, W. D. Kou, and H. Li, "Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol," *Inf. Sci.*, vol. 180, no. 23, pp. 4672–4684, 2010.
- [4] P. J. Zheng and J. W. Huang, "Walsh-Hadamard transform in the homo-morphic encrypted domain and its application in image watermarking," in *Proc. 14th Int. Hiding Conf.*, Berkeley, CA, USA, 2012, pp. 1–15.
- [5] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," *Proc. SPIE*, vol. 6819, pp. 68191E-1–68191E-9, Jan. 2008.
- [6] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.
- [7] W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.
- [8] X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.
- [9] K. D. Ma, W. M. Zhang, X. F. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 553–562, Mar. 2013.
- [10] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 703–716, Jun. 2012.
- [11] S. G. Lian, Z. X. Liu, and Z. Ren, "Commutative encryption and watermarking in video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 774–778, Jun. 2007.
- [12] S. W. Park and S. U. Shin, "Combined scheme of encryption and watermarking in H.264/scalable video coding (SVC)," *New Directions Intell. Interact. Multimedia*, vol. 142, no. 1, pp. 351–361, 2008.
- [13] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [14] S. G. Lian, Z. X. Liu, Z. Ren, and H. L. Wang, "Secure advanced video coding based on selective encryption algorithms," *IEEE Trans. Consumer Electron.*, vol. 52, no. 2, pp. 621–629, May 2006.
- [15] Z. Shahid, M. Chaumont, and W. 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.*, vol. 21, no. 5, pp. 565–576, May 2011.
- [16] M. N. Asghar and M. Ghanbari, "An efficient security system for CABAC bin-strings of H.264/SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 3, pp. 425–437, Mar. 2013.
- [17] T. Stutz and A. Uhl, "A survey of H.264 AVC/SVC encryption," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 325–339, Mar. 2012.
- [18] *Advanced Video Coding for Generic Audiovisual Services*, ITU, Geneva, Switzerland, Mar. 2005.
- [19] J. G. Jiang, Y. Liu, Z. P. Su, G. Zhang, and S. Xing, "An improved selective encryption for H.264 video based on intra prediction mode scrambling," *J. Multimedia*, vol. 5, no. 5, pp. 464–472, 2010.
- [20] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Hoboken, NJ, USA: Wiley, 2003.
- [21] D. K. Zou and J. A. Bloom, "H.264 stream replacement watermarking with CABAC encoding," in *Proc. IEEE ICME*, Singapore, Jul. 2010, pp. 117–121.
- [22] D. W. Xu and R. D. Wang, "Watermarking in H.264/AVC compressed domain using Exp-Golomb code words mapping," *Opt. Eng.*, vol. 50, no. 9, p. 097402, 2011.
- [23] D. W. Xu, R. D. Wang, and J. C. Wang, "Prediction mode modulated data-hiding algorithm for H.264/AVC," *J. Real-Time Image Process.*, vol. 7, no. 4, pp. 205–214, 2012.
- [24] T. Shanableh, "Data hiding in MPEG video files using multivariate regression and flexible macroblock ordering," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 455–464, Apr. 2012.