Reversible Data Hiding in Encrypted H.264/AVC Video Streams

Dawen Xu[1,2(✉)], Rangding Wang[3], and Yun Qing Shi[4]

[1] School of Electronics and Information Engineering, Ningbo University of Technology, Ningbo 315016, China
dawen.xu@njit.edu
[2] Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, Shanghai 200240, China
[3] CKC Software Lab, Ningbo University, Ningbo 315211, China
[4] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982, USA
shi@njit.edu

Abstract. Reversible data hiding in the encrypted domain is an emerging technology because of the privacy-preserving requirements from cloud data management. In this paper, a reversible data hiding scheme in encrypted H.264/AVC video streams is proposed. During H.264/AVC encoding, the intra-prediction mode (IPM), motion vector difference (MVD), and residue coefficients' signs are encrypted using a standard stream cipher. Then, the data-hider, who does not know the original video content, may reversibly embed secret data into the encrypted H.264/AVC video based on histogram-shifting of residue coefficients. With an encrypted video containing hidden data, data extraction can be carried out either in encrypted or decrypted domain. In addition, real reversibility is realized, that is, data extraction and video recovery are free of any error. Experimental results demonstrate the feasibility and efficiency of the proposed scheme.

Keywords: Video encryption · Reversible data hiding · Privacy protection · H.264/AVC

1 Introduction

Since H.264/AVC is the most widely-used video compression standard, the necessity of practical security and privacy preservation for H.264/AVC is unquestionable [1]. To address this problem, H.264/AVC video stream should be distributed over an open network in encrypted format. In some application scenarios, it is necessary to embed some additional message, such as authentication data or media notation, within the encrypted video even though the original video content is unrevealed. With the additional message, the server can manage the video or verify its integrity without having the knowledge of the original content, and thus the privacy is protected. 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, especially for cloud computing [2].

Y.Q. Shi et al. (Eds.): IWDW 2013, LNCS 8389, pp. 141–152, 2014.
DOI: 10.1007/978-3-662-43886-2_10,

However, it is a big challenge to embed hidden data into these encrypted video streams as the compression process would remove redundant information and the encryption process would randomize the compressed bit stream. Till now, few schemes have been appeared in the open literature. In [3], during H.264/AVC compression, the intra-prediction mode (IPM), motion vector difference (MVD) and the first 8 coefficients' signs in each 4×4 block are encrypted, while DCT coefficients' amplitudes are watermarked adaptively. Another commutative watermarking and encryption scheme for MPEG-2 video is proposed in [4], the DCs in intra macroblocks are encrypted or watermarked based on random module addition, while the DCs in other macroblocks and all the ACs' signs are encrypted with a stream cipher or block cipher. However, data embedding is not reversible within the aforementioned schemes. This is critical drawback in some applications. For example, the cloud service provider has no right to introduce permanent distortion due to data hiding in encrypted video. To solve this problem, reversible data hiding technique that completely preserves the video quality in encrypted domain is preferred.

Several reversible data hiding in encrypted image are investigated in [5–8]. In [5], Zhang divided the encrypted image into blocks, and each block carries one bit by flipping 3 Least Significant Bits of each encrypted pixel in a set. Hong et al. [6] improved Zhang's method [5], by adopting new smooth evaluation function and side-match mechanism to decrease the error rate of extracted bits. Later, Zhang further proposed a separable reversible data hiding in encrypted image [7]. In [8], Ma et al. proposed a reversible data hiding in encrypted image by reserving room before encryption. Unfortunately, these algorithms in encrypted image cannot be applied to H.264/AVC stream due to the impact of the coding. In [9], a combined scheme of encryption and watermarking to provide the access right and the authentication of the video content simultaneously is presented. Similarly, the IPMs of 4 × 4 luminance block, the sign bits of texture, and the sign bits of MVDs are encrypted, while IPM is used for reversible watermarking. However, the marked stream is not fully compatible with the specification of H.264 standards, such that a standard decoder which cannot parse a watermarked stream may crash.

In this paper, we propose an algorithm to reversibly embed secret data directly in encrypted H.264/AVC stream. The encrypted H.264/AVC streams are obtained by encrypting the IPM, MVD, and DCT coefficients' signs with a stream cipher. Reversible data hiding in encrypted domain is then performed based on histogram-shifting of residue coefficients. The rest of the paper is organized as follows. In Sect. 2, we describe the proposed scheme. Experimental results are presented in Sect. 3. Finally in Sect. 4, conclusion and discussion are drawn.

2 Proposed Scheme

In this section, a reversible data hiding method in encrypted version of H.264/AVC videos is illustrated, which is made up of video encryption, data embedding, data extraction and video recovery phases.

2.1 H.264/AVC Video Encryption

A. IPM Encryption

Intra-prediction modes in H.264/AVC encoder include Intra_4 × 4, Intra_16 × 16, Intra_chroma and I_PCM [10]. In H.264/AVC, the IPM of Intra_16 × 16 is specified in the mb_type field which also specifies other parameters about this block such as coded block pattern (CBP). Table 1 is the list of mb_type values with their meanings which taken from the standard [11]. In H.264/AVC baseline profile, the mb_type is encoded with the Exp-Golomb code. In order to keep the encrypted streaming format compliance to the H.264/AVC standard decoder, we can encrypt the IPM without modifying the CBP. From Table 1, it can be seen that the combination of CBP is the same in every four lines, and the codeword has the same length respectively in every two lines. Thus for Intra_16 × 16 block, the IPM encryption can be performed by XORing the last bit of the codewords with pseudo-random bits to keep the value of CBP and the length of codeword unchanged which may result in a compliant bit-stream. Pseudo-random bits are determined by an encryption key E_Key1 using a standard stream cipher (e.g., RC4).

Table 1. Macroblock types for I slices and variable length of codeword

Mb type	Name of mb_type	Intra16 × 16 PredMode	Chroma CBP	Luma CBP	Codeword
1	I_16 × 16_0_0_0	0	0	0	01**0**
2	I_16 × 16_1_0_0	1	0	0	01**1**
3	I_16 × 16_2_0_0	2	0	0	0010**0**
4	I_16 × 16_3_0_0	3	0	0	0010**1**
5	I_16 × 16_0_1_0	0	1	0	0011**0**
6	I_16 × 16_1_1_0	1	1	0	0011**1**
7	I_16 × 16_2_1_0	2	1	0	000100**0**
8	I_16 × 16_3_1_0	3	1	0	000100**1**
9	I_16 × 16_0_2_0	0	2	0	000101**0**
10	I_16 × 16_1_2_0	1	2	0	000101**1**
11	I_16 × 16_2_2_0	2	2	0	000110**0**
12	I_16 × 16_3_2_0	3	2	0	000110**1**
13	I_16 × 16_0_0_1	0	0	15	000111**0**
14	I_16 × 16_1_0_1	1	0	15	000111**1**
15	I_16 × 16_2_0_1	2	0	15	00001000**0**
16	I_16 × 16_3_0_1	3	0	15	00001000**1**
17	I_16 × 16_0_1_1	0	1	15	00001001**0**
18	I_16 × 16_1_1_1	1	1	15	00001001**1**
19	I_16 × 16_2_1_1	2	1	15	00001010**0**
20	I_16 × 16_3_1_1	3	1	15	00001010**1**
21	I_16 × 16_0_2_1	0	2	15	00001011**0**
22	I_16 × 16_1_2_1	1	2	15	00001011**1**
23	I_16 × 16_2_2_1	2	2	15	00001100**0**
24	I_16 × 16_3_2_1	3	2	15	00001100**1**

In H.264/AVC, Intra_4 × 4 luminance block has 9 prediction modes, namely mode 0- mode 8. To efficiently compress the prediction mode bits, predictive coding is used to signal Intra_4 × 4 prediction modes [10]. For each current block E, the most probable mode (MPM_E) is estimated from the spatially adjacent upper and left blocks. If the prediction mode of current block is equal to MPM_E, only one bit is needed to signal the prediction mode. In this case, the IPMs are kept unchanged. Otherwise, eight values are required (0 to 7) to signal the prediction mode. The codeword is composed of one sign bit "0" and three bits fixed-length code. The IPM encryption is performed by XORing three bits fixed-length codeword B with pseudo-random bits which are determined by an encryption key E_Key2 using a standard stream cipher as shown in Fig. 1. Obviously, the length of the bit string corresponding to the encrypted IPM is the same as that corresponding to the original IPM. In summary, IPM encryption implies changing the actual mode to another one without violating the semantics and bitstream compliance.

B. MVD Encryption

In H.264/AVC baseline profile, Exp-Golomb entropy coding is used to encode MVD. The codeword of Exp-Golomb is constructed as [*Mzeros*][1][*INFO*] [10]. *M_zeros* and *INFO* are an M leading zero bits before the first bit 1 and the following M-bit field carrying information, respectively. The length of each Exp-Golomb codeword is $(2M - 1)$. Each codeword can be constructed by the encoder based on its index *code_num*:

$$\begin{cases} M = floor(\log_2(code\_num + 1)) \\ INFO = code\_num + 1 - 2^M \end{cases} \tag{1}$$

The MVD k to be encoded is mapped to *code_num* as follows:

$$code\_num = \begin{cases} 2|k| & k \leq 0 \\ 2|k| - 1 & k > 0 \end{cases} \tag{2}$$

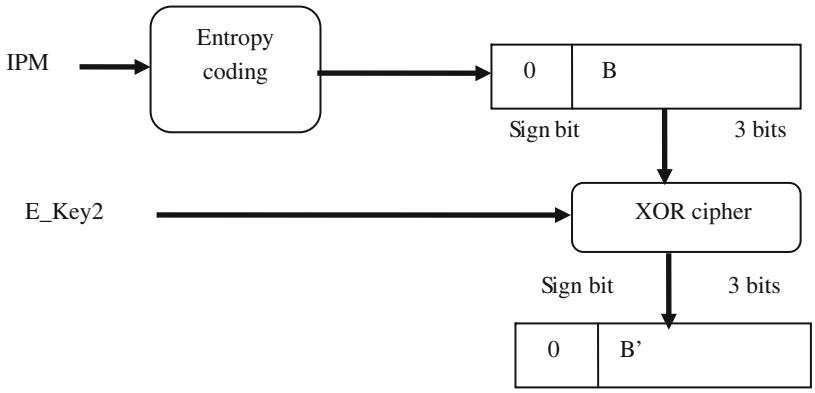


Fig. 1. IPM encryption process for Intra_4 × 4 block

Table 2. MVD and corresponding Exp-Golomb codeword

MVD	*code_ num*	*Codeword*
0	0	1
1	1	010
−1	2	011
2	3	00100
−2	4	00101
3	5	00110
−3	6	00111
4	7	0001000
−4	8	0001001
5	9	0001010
−5	10	0001011
…	…	…

Table 2 shows MVD and corresponding Exp-Golomb codeword. In order to avoid the bit-overhead and satisfy the format compliance, only sign bits of MVDs are encrypted using a standard stream cipher with an encryption key E_Key3. For example, the codewords corresponding to "2" and "−2" are "00100" and "00101", respectively, which have the same length.

C. Residue Encryption
In H.264/AVC baseline profile, Context-Adaptive Variable Length Coding (CAVLC) is used to encode the quantized transform coefficients of a residual block. The signs ("0"—positive, "1"–negative) of non-zero coefficients are encrypted using a standard stream cipher with an encryption key E_Key3. In this method, the luminance residue coefficients are encrypted, while the chroma residue coefficients keep unchanged. The bit-overhead may be generated after an encryption due to the CAVLC entropy coding. However, the bit rate of encrypted formats is very close to that of unencrypted formats. This will be confirmed in the experimental results.

2.2 Data Embedding in Encrypted Video

Once the data hider receives the encrypted video, he can embed some information into it for the purpose of media notation or integrity authentication. Since only the coefficients' signs are encrypted, a reversible data hiding method based on histogram-shifting of residue coefficients is proposed.

A. Embedding Zone Selection
The embedding space can be created by modifying the histogram of the residual coefficients. First, find the two highest bins in the histogram (excluding zero coefficients) denoted by T_p and T_n, respectively. For simplicity, we select $T_p = 1$ and $T_n =- 1$.

In the histogram shifting approach, all coefficients lying between the range of peak and zero points are shifted with one level. When the number of modified coefficients is

increased, the quality of host video will be affected. Since the DC and low frequency coefficients contain most of the energy and embedding data in these may affect the video quality and the bit-rate significantly, in order to imperceptibly embed a certain amount of message, we choose mid- and high-frequency DCT residual coefficients to embed data. Thus the number of modified coefficients is decreased since only those coefficients located in the mid- and high-frequency are modified. All of the transform coefficients within the 4×4 block can be marked in a sequential order, from 0, 1, …, up to 15 in the zig-zag order. Suppose the embedding region is $R = [T_1, T_2]$, where T_1 and T_2 are two sequential numbers along the zig-zag scan order, i.e., $0 \leq T_1 < T_2 \leq 15$. That is, the transform coefficients whose values are between $[T_1, T_2]$ are modified to embed data while those outside the range are left unchanged. After some experimental investigation, we set $T_2 = 15$. Since coefficients vary widely for different blocks, the threshold T_1 is dynamically adopted according to the DC value of the current 4×4 block, which is denoted as DC_{cur}. Of course, threshold T_1 may still be improved to achieve optimization.

$$T_1 = \begin{cases} 3 & if\ DC_{cur} < 1 \\ 4 & if\ 1 \leq DC_{cur} \leq 5 \\ 5 & if\ 6 \leq DC_{cur} \leq 10 \\ 6 & if\ 11 \leq DC_{cur} \leq 20 \\ 7 & if\ DC_{cur} \geq 21 \end{cases} \tag{3}$$

Data embedding is performed in those 4×4 blocks in P-frames only when the DC or the first AC coefficients are nonzero, since I-frames are crucial for video signal. Our simulation results demonstrate that we can embed the additional data with a large capacity in P-frames while preserving high visual quality.

B. Data Embedding

Many reversible data hiding methods have been proposed, such as the methods based on difference expansion and histogram shifting. Histogram shifting is one of the most popular methods which modify the histogram in such a way that certain bins are shifted to create vacant space while some other bins are utilized to carry data by filling the vacant space. Before data embedding, partial decoding is performed on the encrypted stream to obtain the residue coefficients, which denote as $f_{i,j}(k)$. Here, $f_{i,j}(k)$ is encrypted residue, that is, the data hider does not have the key to decrypt and get the plain values. Data hiding is performed directly in encrypted domain by modifying the encrypted coefficients based on histogram shifting. The stego encrypted coefficients are then replaced back and all the coefficients are CAVLC coded.

Different from the traditional reversible data hiding method [12], in this method, the bins 1 and -1 are utilized for data embedding and other bins (except bin 0) for shifting. Suppose the to-be-embedded message is a binary signal denoted as $W = \{w_l | l = 1, 2, \cdots, M, w_l \in \{0, 1\}\}$. Our data hiding algorithm in the encrypted domain can be described as follows.

Case 1: If $f_{i,j}(k) > 1$ or $f_{i,j}(k) < -1$, shift $f_{i,j}(k)$ by 1 units to the right and left, respectively. That is,

$$\overline{f_{i,j}}(k) = \begin{cases} f_{i,j}(k) - 1, & if\ f_{i,j}(k) < -1 \\ f_{i,j}(k) + 1, & if\ f_{i,j}(k) > 1 \end{cases} \tag{4}$$

where $f_{i,j}(k)$ is the k-th original quantized DCT coefficient in the j-th block of the i-th macroblock, $\overline{f_{i,j}}(k)$ is the corresponding coefficients with hidden data.

Case 2: If $f_{i,j}(k) = 1$ or $f_{i,j}(k) = -1$, modify $f_{i,j}(k)$ according to the message bit w_l to be embedded.

$$\overline{f_{i,j}}(k) = \begin{cases} f_{i,j}(k) - 1, & if\ f_{i,j}(k) = -1,\ and\ w_l = 1 \\ f_{i,j}(k) & if\ f_{i,j}(k) = -1,\ and\ w_l = 0 \\ f_{i,j}(k) + 1 & if\ f_{i,j}(k) = 1,\ \ and\ w_l = 1 \\ f_{i,j}(k) & if\ f_{i,j}(k) = 1,\ \ and\ w_l = 0 \end{cases} \tag{5}$$

Case 3: If $f_{i,j}(k) = 0$, $\overline{f_{i,j}}(k) = f_{i,j}(k)$, i.e., $f_{i,j}(k)$ keeps unchanged, i.e., the marked coefficient $\overline{f_{i,j}}(k)$ is taken as $f_{i,j}(k)$ itself. An example on how to embed data by histogram shifting is illustrated in Fig. 2.

2.3 Data Extraction and Original Video Recovery

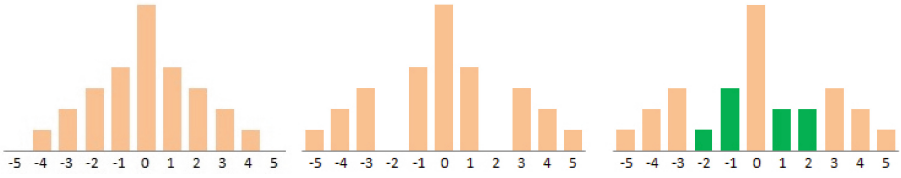
In this scheme, the hidden data can be extracted either in encrypted or decrypted domain. Besides, our method is also reversible, where the hidden data could be removed to obtain the original video.

A. Scheme I: Encrypted Domain Extraction
From the stego encrypted video, the embedded data bit w_l can be extracted as

$$\bar{w}_l = \begin{cases} 0 & if\ \overline{f_{i,j}}(k) = \pm 1 \\ 1 & if\ \overline{f_{i,j}}(k) = \pm 2 \end{cases} \tag{6}$$

The original encrypted residue coefficients $f_{i,j}(k)$ can be recovered as

$$f_{i,j}(k) = \begin{cases} \overline{f_{i,j}}(k), & if\ \overline{f_{i,j}}(k) \in \{0, \pm 1\} \\ \overline{f_{i,j}}(k) - 1, & if\ \overline{f_{i,j}}(k) \geq 2 \\ \overline{f_{i,j}}(k) + 1, & if\ \overline{f_{i,j}}(k) \leq -2 \end{cases} \tag{7}$$



(a) original histogram (b) shifted histogram (c) stego histogram

Fig. 2. An example of the histogram-shifting method

Fig. 3. Original video frames

Since the whole process is entirely performed in the encrypted domain, it avoids the leakage of original content. With the encryption keys, the content owner can further decrypt the video to get the original cover video.

B. Scheme II: Decrypted Domain Extraction

In some cases, users want to decrypt the video first and then extract the hidden data from the decrypted video. For example, an authorized user, which owned the encryption key, decrypted the encrypted video with hidden data. The decrypted video still includes the hidden data, which can be used to trace the source of the data. The whole process of decryption, data extraction and video recovery is as follows.

Step1: Generate the stego decrypted video with the encryption keys. Because of the symmetry of the XOR operation, the decryption operation is symmetric to the encryption operation. That is, the encrypted data can be decrypted by performing XOR operation with generated pseudorandom bits, and then two XOR operations cancel each other out, which renders the original plain text. Since the pseudorandom bits depend on the encryption keys, the decryption is possible only for the authorized users.

Step2: The embedded data bit w_l can also be extracted from the stego decrypted video using Eq. (6), since the encryption simply changed the signs of the coefficients. For example, the coefficient value "1" may be changed to "−1" during the encryption process, but "1" and "−1" are all correspond to the hidden bit "0". The difference is that $f_{i,j}(k)$ should be decrypted coefficients.

Step3: The original video can be further recovered via Eq. (7). Similarly, $f_{i,j}(k)$ should be decrypted coefficients.

3 Experimental Results

The proposed data hiding scheme has been implemented in the H.264/AVC JM-12.2 reference software [13]. Four well-known standard video sequences (i.e., *Carphone*, *News*, *Foreman*, *and Hall*) in QCIF format (176 × 144) at the frame rate 30 frames/s are used for our simulation. The first 100 frames in each video sequence are used in the experiments. The GOP (Group of Pictures) structure is "IPPPP: one I frame followed four P frames (QP: I 28, P 28)".

A. Security of the encryption algorithm

For a video encryption scheme, the security depends on cryptographic security and perceptual security. Cryptographic security denotes the security against cryptographic

Fig. 4. Encrypted video frames

attacks, which depends on the cipher. In the proposed scheme, the secure stream cipher is used to encrypt the bitstream and the secret data, which has been proved its security against attack. Perceptual security refers to the encrypted video is unintelligible. The proposed scheme encrypts IPM, MVD and residue coefficients, which keeps the encrypted video unintelligible. The encryption results are shown in Fig. 4. In general, scrambling performance of the described encryption system is more than adequate.

B. Stego Video Quality

An original frame from each video is shown in Fig. 3. The decrypted video frames with hidden data are given in Fig. 5. The degradation of the decoded video quality should be maintained at an acceptable range, whether or not the hidden data is removed. Since the embedding scheme is reversible, the original cover content can be perfectly recovered after extraction of the hidden data. At the same time, in the experiments, no visible artifacts can be observed in all of the decrypted videos containing hidden data.

Besides subjective observation, *PSNR* (Peak Signal to Noise Ratio), *SSIM* (Structural Similarity Index), and VQM (Video Quality Measurement) are adopted to evaluate the perceptual quality [14, 15]. The SSIM index lies in the range between 0 and 1, where 0 indicates zero correlation, i.e. the reference image is entirely different than the target, and 1 indicates that they are identical. Since H.264/AVC is lossy compression, in order to better illustrate the data hiding on the video quality, the visual quality of non-stego video stream should be tested. Video sequence after compression and decompression process is used as the target, while the original uncompressed video sequence is used as the reference video clip. Similarly, in order to test the visual quality of stego video stream, video sequence after compression, encryption, data hiding, decryption, and decompression process is used as the target.

Fig. 5. Decrypted video frames with hidden data

Table 3. Embedding capacity, *PSNR*, *SSIM* and *VQM* in directly decrypted videos

Sequence	*QP*	*Maximum capacity*	*BR_var* (%)	*PSNR* (dB)		*SSIM*		*VQM*	
				Non-stego	*Stego*	*Non-stego*	*Stego*	*Non-stego*	*Stego*
Carphone	24	16468	2.89	40.94	40.36	0.9813	0.9806	0.5703	0.5806
	28	5194	1.71	37.95	37.75	0.9697	0.9693	0.7284	0.7330
	32	1339	0.75	35.07	35.00	0.9529	0.9527	0.9019	0.9054
News	24	10559	1.88	40.82	40.37	0.9848	0.9842	0.5847	0.5925
	28	4597	1.33	37.78	37.57	0.9727	0.9723	0.7745	0.7790
	32	1686	0.79	34.57	34.48	0.9531	0.9527	1.0064	1.0090
Foreman	24	18681	2.63	39.17	38.73	0.9726	0.9720	0.6168	0.6284
	28	4689	1.30	36.36	36.22	0.9574	0.9571	0.7926	0.7983
	32	1070	0.53	33.66	33.61	0.9372	0.9371	0.9873	0.9908
Hall	24	11255	2.17	40.33	39.86	0.9744	0.9740	0.6536	0.6627
	28	4928	1.62	37.92	37.66	0.9658	0.9655	0.7848	0.7924
	32	2098	1.13	34.98	34.86	0.9527	0.9524	0.9542	0.9590

That is, in this case, the target video containing the hidden data. VQM presents another approach in video quality measuring that correlates more with the Human Visual System (HVS). In general, the lower VQM value indicates higher perceptual video quality, and zero indicates excellent quality. The experimental results are shown in Table 3. We can see that there is only a very small change in PSNR, SSIM and VQM values. It is almost impossible to detect the degradation in video quality caused by data hiding.

C. Embedding Capacity

The maximum payload capacity in each video encoded with different *QP* (quantization parameter) values is given in Table 3. Payload of the proposed scheme depends on type of video content and the QP values. The reason for this is that each video stream has a different number of qualified coefficients. For example, when *QP* value equals to 24, a large number of quantized coefficients can be used for data hiding and hence payload is high. While *QP* value equals to 32, the payload is lower, since quantized coefficients which can be used for data hiding are fewer. Higher payload can be attained if data hiding is allowed in I-frames, but it will have more effect on visual quality of video.

D. Bit Rate Variation

To further evaluate the performance of the proposed scheme, bit rate variation caused by encryption and data hiding is also introduced [16].

$$BR\_\text{var} = \frac{BR\_em - BR\_orig}{BR\_orig} \times 100\% \tag{8}$$

where *BR_em* is the bit rate generated by encryption and data embedding encoder, and *BR_orig* is the bit rate generated by the original encoder. The results of *BR_var* are

also depicted in Table 3. Changing zero-quantized residuals to nonzero values in course of embedding can significantly increase the video bit rate when coefficients are encoded using CAVLC codes. However, zero-quantized residuals are not modified in our scheme, so bit rate is increased, but still below 2.89 %. This effect can be almost ignored.

4 Conclusion and Discussion

In this paper, an algorithm to reversibly embed secret data in encrypted H.264/AVC streams is presented, which consists of video encryption, data embedding and data extraction three phases. The IPM, MVD, and the sign bits of residue coefficients are encrypted using a standard stream cipher without violating format compliance. The data-hider can embed the secret data into the encrypted video using the traditional histogram-shifting method, even though he does not know the original video content. Data extraction is separable from the video decryption. In other words, the additional data can be extracted either in encrypted domain or decrypted domain. Furthermore, this algorithm can achieve real reversibility, and high quality of marked decrypted videos. One of the possible applications of this method is video annotation in cloud computing where high video quality and reversibility are greatly desired.

Although reversible data hiding and cryptography have reached a certain degree of maturity, but reversible data hiding in encrypted domain is a highly interdisciplinary area of research. Technical research in this field has only just begun, and there is still an open space for research in this interdisciplinary research area.

Acknowledgements. The first author gratefully acknowledges the support of K.C. Wong Education, Hong Kong. This work is also supported by the National Natural Science Foundation of China (61301247), Zhejiang Provincial Natural Science Foundation of China (LY13F020013, LY12F01001), Ningbo Natural Science Foundation (2013A610059, 2011A610182) and the Opening Project of Shanghai Key Laboratory of Integrate Administration Technologies for Information Security (AGK2013004).

References

1. Stutz, T., Uhl, A.: A survey of H.264 AVC/SVC encryption. IEEE Trans. Circ. Syst. Video Technol. **22**(3), 325–339 (2012)
2. Lu, W.J., Varna, A., Wu, M.: Secure video processing: problems and challenges. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011), Prague, Czech Republic, pp. 5856–5859 (2011)
3. Lian, S.G., Liu, Z.X., Ren, Z.: Commutative encryption and watermarking in video compression. IEEE Trans. Circ. Syst. Video Technol. **17**(6), 774–778 (2007)
4. Lian, S.G.: Quasi-commutative watermarking and encryption for secure media content distribution. Multimedia Tools Appl. **43**, 91–107 (2009)
5. Zhang, X.P.: Reversible data hiding in encrypted image. IEEE Signal Process. Lett. **18**(4), 255–258 (2011)

6. Hong, W., Chen, T.S., Wu, H.Y.: An improved reversible data hiding in encrypted images using side match. IEEE Signal Process. Lett. **19**(4), 199–202 (2012)
7. Zhang, X.P.: Separable reversible data hiding in encrypted image. IEEE Trans. Inf. Forensics Secur. **7**(2), 826–832 (2012)
8. Ma, K.D., Zhang, W.M., Zhao, X.F., et al.: Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans. Inf. Forensics Secur. **8**(3), 553–562 (2013)
9. Park, S.W., Shin, S.U.: Combined scheme of encryption and watermarking in H.264/ scalable video coding (SVC). In: New Directions in Intelligent Interactive Multimedia, vol. 142, pp. 351–361 (2008)
10. Richardson, I.E.G.: H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia. Wiley, Chichester (2003)
11. ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services (2005)
12. Ni, Z.C., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Trans. Circ. Syst. Video Technol. **16**(3), 354–362 (2006)
13. H.264/AVC Reference Software JM 12.2. http://iphome.hhi.de/suehring/tml/
14. Xu, D.W., Wang, R.D., Wang, J.C.: Prediction mode modulated data-hiding algorithm for H.264/AVC. J. Real-Time Image Proc. **7**(4), 205–214 (2012)
15. Sun, T.F., Jiang, X.H., Lin, Z.G., Wang, S.L.: An H.264/AVC video watermarking scheme in VLC domain for content authentication. China Commun. **7**(6), 30–36 (2010)
16. Xu, D.W., Wang, R.D., Wang, J.C.: A novel watermarking scheme for H.264 AVC video authentication. Sig. Process. Image Commun. **26**(6), 267–279 (2011)