Received: 3 September 2018 | Revised: 13 February 2019 | Accepted: 1 October 2019

DOI: 10.4218/etrij.2018-0484

ORIGINAL ARTICLE

ETRI Journal WILEY

Data hiding in partially encrypted HEVC video

Dawen Xu

School of Electronics and Information Engineering, Ningbo University of Technology, Ningbo, China

Correspondence
Dawen Xu, School of Electronics and Information Engineering, Ningbo University of Technology, Ningbo, China.
Email: dawenxu@126.com

Funding information
This research was supported by the National Natural Science Foundation of China (61771270), Zhejiang Provincial Natural Science Foundation of China (LY17F020013, LR20F020001), and Ningbo Natural Science Foundation (2018A610054).

In this study, an efficient scheme for hiding data directly in partially encrypted versions of high efficiency video coding (HEVC) videos is proposed. The content owner uses stream cipher to selectively encrypt some HEVC-CABAC bin strings in a format-compliant manner. Then, the data hider embeds the secret message into the encrypted HEVC videos using the specific coefficient modification technique. Consequently, it can be used in third-party computing environments (more generally, cloud computing). For security and privacy purposes, service providers cannot access the visual content of the host video. As the coefficient is only slightly modified, the quality of the decrypted video is satisfactory. The encrypted and marked bitstreams meet the requirements of format compatibility, and have the same bit rate. At the receiving end, data extraction can be performed in the encrypted domain or decrypted domain that can be adapted to different application scenarios. Several standard video sequences with different resolutions and contents have been used for experimental evaluation.

KEYWORDS

context adaptive binary arithmetic coding (CABAC), data hiding, encrypted domain, high efficiency video coding (HEVC)

1 | INTRODUCTION

With the rapid popularization and application of mobile intelligent terminals, video data are growing explosively. Cloud computing provides the most effective solution for the storage and processing of large videos. However, under the existing cloud computing architecture, users do not have absolute control of data privacy. Consequently, security and privacy protection have become major concerns of the public [1]. An effective way to ensure security and confidentiality is for the video provider to encrypt the original video before uploading it to the cloud. All the processing and computing in the cloud are performed directly in the encrypted domain, following which the results are provided to the users. An authorized user with the decryption key can obtain plaintext data after decryption. However, video data lose its original nature after being encrypted. Thus, effectively managing massive ciphertext videos in cloud, and protecting their integrity and reliability have become a pressing problem. For the purpose of video management, tampering detection, or ownership declaration, embedding some additional information directly into the encrypted video is more promising.

Over the past few years, some substantial works have been conducted on data hiding in the encrypted domain. Earlier, Zhao and others [2] proposed a watermarking scheme with flexible watermarking capacity in the encrypted domain. Then, Guo and others [3] proposed an efficient watermarking scheme based on the Paillier cryptosystem that is robust against privacy attacks. In these schemes [2,3], homomorphic encryption (eg, Paillier cryptosystem) is utilized to encrypt the original media. However, the main problem with current homomorphic encryption is that it will result in data expansion and high overhead. To circumvent this shortcoming, Subramanyam and others [4] proposed a robust watermarking algorithm for compressed and encrypted

JPEG2000 images by exploiting their homomorphic property. In addition, Liu and others [5] proposed a commutative encryption and watermarking scheme based on compressive sensing that is robust against noise attacks during transmission. Xiang and He [6] proposed an efficient database authentication watermarking scheme in the encrypted domain. Rad and others [7] proposed a unified data embedding and scrambling method for achieving high payload and adaptive scalable quality degradation. It should be noted that reversible data hiding in the encrypted domain [8–12] that ensures the accuracy of reconstructed images has also recently become a research hotspot.

The above algorithms are all designed for digital images. Because video codec is much more complicated than the process of image compression, these algorithms cannot be directly applied to digital video. Fortunately, some algorithms have been proposed for hiding data in encrypted H.264/AVC videos. In [13], an effective algorithm is designed to implement commutative encryption and watermarking during the H.264/AVC video coding process. The signs of the motion vector difference (MVD), intra-prediction mode (IPM), and discrete cosine transform (DCT) coefficients' are encrypted, while the DCT coefficients' amplitudes are watermarked. The encryption and watermarking operations are commutative. In our previous works [14,15], some techniques of hiding data directly in the encrypted H.264/AVC stream are presented. By mining the structure of the H.264/AVC bitstream, the code words of the three important syntax elements (ie, IPMs, MVDs, and DCT coefficients) are efficiently encrypted using stream ciphers. The data hider can embed the secret message in encrypted videos using specific code words substitution technology. The methods in [14,15] are based on the context-adaptive variable-length coding (*CAVLC*) entropy coding. In addition to *CAVLC*, *context-adaptive binary arithmetic coding* (*CABAC*) is also supported in H.264/AVC, which demonstrates higher compression performance. Therefore, we subsequently proposed some data hiding methods for encrypted H.264/AVC videos based on the CABAC bin-string substitution [16,17]. It should be noted that reversible data hiding in encrypted H.264/AVC videos were designed in [18,19].

As a new standard for video compression, high efficiency video coding (HEVC) has the potential to outperform earlier standards such as H.264/AVC. Consequently, it is necessary to develop an efficient data hiding scheme for encrypted HEVC video. However, in reality, the literature is very scant in this regard. A reversible data hiding and encryption scheme with separability is proposed for HEVC videos in [20]. However, the compression ratio will increase after data embedding, because the data hiding operation increases the magnitudes of the nonzero coefficients. Ideally, the HEVC encryption and data embedding technique should retain the video's format compliance and amount of bit rate. Based on the above situation, we propose an efficient scheme to embed secret messages directly in partially encrypted HEVC streams. Since arithmetic coding is extremely sensitive to bit errors, we can selectively encrypt binary strings instead of bitstreams by analyzing the properties of the CABAC. Subsequently, data hiding is performed in the encrypted domain using the coefficient modification technique, thereby preserving the confidentiality of the content. In this way, the overall bit rate and format compliance of the encrypted and marked video remain completely unchanged. These properties have been verified through extensive experiments.

The remainder of this article is structured as follows. Section 2 describes the details of the proposed scheme including the selective encryption of HEVC videos, data embedding, and data extraction. Experimental results and performance analysis are provided in Section 3. Finally, conclusions are drawn, and recommendation for future work is presented in Section 4.

2 | PROPOSED SCHEME

The proposed scheme consists of three parts, namely selective encryption of HEVC video, data embedding in encrypted HEVC videos, and data extraction. First, the content owner encrypts the original HEVC bitstream using stream ciphers before sending it to the data hider. Then, the data hider embeds some secret messages into the encrypted video using the *coefficient modulation method*. It is worth noting that the data hider has no right to access the original video content. On side of the receiver, data extraction and decryption are completely separable, that is, they can be done in both the encrypted and decrypted domains. The overall framework is shown in Figure 1.

2.1 | Selective encryption of HEVC video

The CABAC is a form of entropy coding used in H.264/AVC and HEVC. The core design of the CABAC involves the key elements of binarization, context modeling, and binary arithmetic coding [21]. Binarization maps the syntax elements to the binary-valued symbols (*bins*) that includes the *k*th order truncated Rice (TRk), *k*th order Exp-Golomb (EGk), and fixed-length (FL) binarization. According to the coding mode decision, a *regular or bypass* coding engine can be selected to further process each bin value in CABAC. The regular mode has a context modeling stage, in which a probability context model is selected. In contrast, the bypass mode assumes an equiprobable model, as illustrated by the lower right branch of the switch in Figure 1.

Recently, CABAC-based encryption has become a key research topic, because encryption should ensure that the

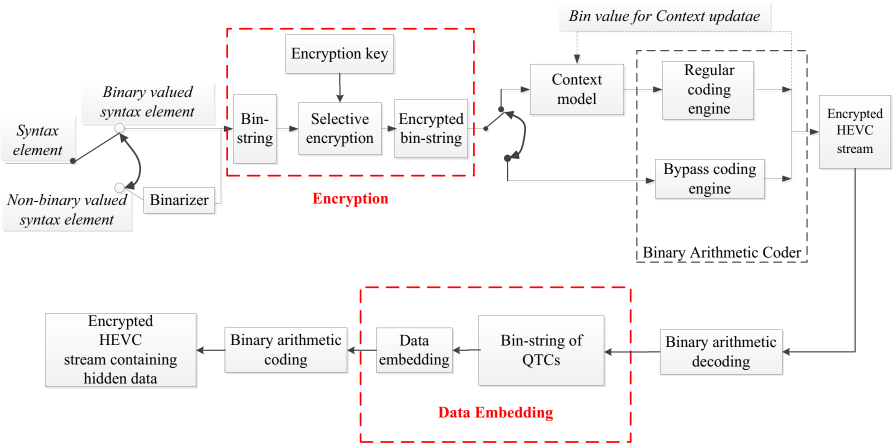


FIGURE 1 Diagram of the proposed scheme

video content is visually distorted, while maintaining its format compliance and bit rate. Generally, selective encryption is performed after the binarization process in the CABAC. Earlier, Shahid and others [22] pointed out that the syntax elements that fulfill the criteria for the encryption of the H.264/AVC-compliant bitstream are the suffixes of nonzero coefficients and their sign bits. However, Wang and others [23] demonstrated that encrypting the suffix cannot produce effective scrambling, and is not as efficient as only encrypting the sign bits of the nonzero coefficients. In [24], luma prediction modes for intra-blocks are included in the encryption space, which can yield superior levels of protection. However, the problem of intra-prediction encryption will decrease the compression efficiency slightly. Recently, Sallam and others [25] proposed an HEVC selective encryption technique to encrypt the coefficients sign bits, the suffixes of the remaining absolute value that are binarized by the Exp-Golomb order zero (EG0), the MVD sign bits and the MVD absolute value suffixes that are binarized by the Exp-Golomb order one (EG1).

As *bypassed bins* do not exploit and update probability models during the arithmetic coding, their encryption is possible as long as the induced modifications do not disrupt a standard decoding process. Considering encoding efficiency and format compliance, the signs of the *quantized transform coefficients* (*QTCs*) and *MVDs* are encrypted in the proposed method. The encrypted bin strings are then coded through the binary arithmetic coding (BAC), as shown in Figure 1. The encrypted bitstream remains HEVC compliant, and has the same bit rate as the original bitstream.

QTC Encryption: In the CABAC, the sign bit and the magnitude of the nonzero QTC are coded separately. The sign bit is encoded in the *bypass* mode, while its magnitude is first binarized through UEG0, and then coded using the BAC. UEG0 is the concatenation of the truncated unary (TU) code and the 0th Exp-Golomb (EG0) code. To maintain format compliance and the bit rate value, the sign bits in both the I-frames and P-frames are encrypted with a stream cipher. The encryption operation is as demonstrated in (1); it involves XORing X_i with K_i to obtain C_i. Here, X_i denotes the sign of the ith nonzero *QTC*. If the *QTC* is positive, the corresponding value of X_i is equal to 0. Otherwise, the corresponding value of X_i is equal to 1. K_i is a random bit generated by the stream cipher, and C_i is the corresponding cipher value.

$$C_i = X_i \oplus K_i. \tag{1}$$

In the HEVC, the sign of each nonzero QTC is encoded in the *bypass* mode. Because the sign bits are *bypassed* without further processing, directly applying encryption to the sign bits has no impact on the compression performance.

MVD Encryption: The motion vectors (*MVs*) play an important role in inter-prediction in the HEVC, and are very important for video reconstruction. Because the motion data of a block are correlated with the neighboring blocks, motion data are not directly coded, but predictively coded, based on the neighboring motion data. In the HEVC, a new tool called the advanced motion vector prediction (AMVP) is used. The MVDs are actually coded in the bitstream. Furthermore, the signs of the MVDs are coded in the *bypass* mode, which means changing zero to one or vice versa will never affect the compression ratio. To maintain the length of the coded stream, the signs of the MVDs are also encrypted by applying the bitwise XOR operation with a stream cipher.

In summary, both the sign bits of the QTCs and the sign bits of the MVDs are encrypted. Its advantages are that it has no impact on the compression rate, and can achieve effective perceptual scrambling. These will be confirmed in the following experimental results. In addition, because the simple bitwise XOR operation only needs to be performed in the encryption process, its computational complexity is very low, and it can thus be used in real-time applications.

2.2 | Data embedded in the encrypted domain by QTC modification

In the HEVC, several modifications are introduced to address large transform blocks and enhance the throughput by encoding more bins in bypass mode [26]. The nonzero QTC level is encoded as varied combinations of three-level indicators (syntax elements) that indicate if the coefficient level is greater than 1 (*coeff_abs_level_greater1_flag*), greater than 2 (*coeff_abs_level_greater2_flag*), and also indicate the remaining actual values (*coeff_abs_level_remaining*). The syntax element *coeff_abs_level_remaining* is coded in the bypass mode to increase the throughput. The HEVC employs Golomb-Rice codes for small values, and switches to the Exp-Golomb code for larger values. The Rice parameter, *m*, is set to 0 at the beginning of each coefficient group, and is conditionally updated depending on the previous value of the parameter and the current absolute level as follows:

$$\text{If } absCoeffLevel > 3 \times 2^{m},\ \ m = \min(4, m+1). \quad (2)$$

Let the *baseLevel* of a coefficient be defined as follows:

$$\begin{aligned} baseLevel = {} & significant\_coeff\_flag \\ & + coeff\_abs\_level\_greater1\_flag \\ & + coeff\_abs\_level\_greater2\_flag, \end{aligned} \quad (3)$$

where a flag has a value of 0 or 1, and is inferred to be 0 if not present. Then, the absolute value of the coefficient is simply denoted as follows:

$$absCoeffLevel = baseLevel + coef\_abs\_level\_remaining. \quad (4)$$

The value of the parameter, *m*, may range from 0 to 4. Bypass coding is specified for the bins of the syntax element, *coeff_abs_level_remaining*, and the binarization of this syntax element is specified as the Golomb-Rice codes and Exp-Golomb codes. Tables 1–5 show the binarization of the remaining level when *m* is 0 to 4.

Data embedding can be accomplished by substituting the eligible bin strings of the *coeff_abs_level_remaining* in Tables 1–5. To enhance the security, a stream cipher is used to encrypt the message according to the data hiding key. The encrypted version can be denoted as $W = \{w(i) | i = 1, 2, ..., K, w(i) \in \{0, 1\}\}$. Generally, the binary bit can be embedded through a QTC modification, as shown in the following equation.

$$\overline{absCoeffLevel} = \begin{cases} absCoeffLevel + w(i) & \text{if } coeff\_abs\_level\_remaining \% 2 = 0, \\ absCoeffLevel - 1 + w(i) & \text{if } coeff\_abs\_level\_remaining \% 2 = 1, \end{cases} \quad (5)$$

TABLE 1 Binarization for *the remaining level* when *m* = 0

coeff_abs_level_ remaining	Prefix	Suffix	Suffix range
0	0		
1	10		
2	110		
3	1110		
4–5	11110	x	0–1
6–9	111110	xx	0–3
10–17	1111110	xxx	0–7
18–33	11111110	xxxx	0–15
…	…	…	…

TABLE 2 Binarization for *the remaining level* when *m* = 1

coeff_abs_level_ remaining	Prefix	Suffix	Suffix range
0–1	0	x	0–1
2–3	10	x	0–1
4–5	110	x	0–1
6–7	1110	x	0–1
8–11	11110	xx	0–3
12–19	111110	xxx	0–7
20–35	1111110	xxxx	0–15
36–67	11111110	xxxxx	0–31
…	…	…	…

TABLE 3 Binarization for *the remaining level* when *m* = 2

coeff_abs_level_remaining	Prefix	Suffix	Suffix range
0–3	0	xx	0–3
4–7	10	xx	0–3
8–11	110	xx	0–3
12–15	1110	xx	0–3
16–23	11110	xxx	0–7
24–39	111110	xxxx	0–15
40–71	1111110	xxxxx	0–31
…	…	…	…

where $\overline{absCoeffLevel}$ denotes the absolute coefficient level following data embedding and % denotes the modulo operation. Because the *absCoeffLevel* and *coeff_abs_level_remaining* are interrelated, the *coeff_abs_level_remaining* will also be changed when the $\overline{absCoeffLevel}$ is modified. Similarly, we can identify it as $\overline{coeff\_abs\_level\_remaining}$. However, it is not possible to modify the QTCs in the following cases:

TABLE 4 Binarization for *the remaining level* when $m = 3$

coeff_abs_level_remaining	Prefix	Suffix	Suffix range
0–7	0	xxx	0–7
8–15	10	xxx	0–7
16–23	110	xxx	0–7
24–31	1110	xxx	0–7
32–47	11110	xxxx	0–15
48–79	111110	xxxxx	0–31
…	1111110	…	…
	…		

TABLE 5 Binarization for *the remaining level* when $m = 4$

coeff_abs_level_remaining	Prefix	Suffix	Suffix range
0–15	0	xxxx	0–15
16–31	10	xxxx	0–15
32–47	110	xxxx	0–15
48–63	1110	xxxx	0–15
64–95	11110	xxxxx	0–31
…	…	…	…

1. When $m = 0$ and *coeff_abs_level_remaining* ≤ 3, it can be seen from Table 1 that no pair of bin strings have the same size. This means that the equal length substitution constraint is not satisfied.
2. When $m = 1$, the rice parameter, m, will be updated when the current *absCoeffLevel* is greater than the threshold, 6. According to (2), if the value of *absCoeffLevel* is changed from 6 to 7 due to data hiding, the new value may fall into a different region. In this case, the wrong Rice parameter m will be used for decoding the *absCoeffLevel*. To satisfy format compatibility, no modification is made when *absCoeffLevel* is 6 or 7.
3. When $m = 2$, the rice parameter, m, will be updated when the current *absCoeffLevel* exceeds the threshold, 12. To maintain format compatibility, no modification is made when *absCoeffLevel* is 12 or 13.
4. When $m = 3$, the rice parameter, m, will also be updated when the current *absCoeffLevel* exceeds the threshold, 24. Similarly, no modification is made when *absCoeffLevel* is 24 or 25.

The advantage of the algorithm is that the data embedding can be done through simple coefficient modification. Its computational complexity is very low. Furthermore, the length of the marked bin string is equal to the length of the original bin string, and thus the bit rate is preserved as well. Another point to note is that the coefficient modification is only performed within the P-frames, whereas all the *bin strings* of the *absCoeffLevel* in the I-frames remain unchanged. The reason for not selecting the I-frames is that the distortion of I-frames due to data hiding will propagate to subsequent P-frames. Generally, the embedding capacity of the P-frames is relatively small, because they are highly compressed using motion compensation and entropy coding.

2.3 | Data extraction

At the receiver's end, the authorized users can extract the hidden data. Similar to the methods in [16,17], the hidden messages can be extracted before or after video decryption. Data extraction in the encrypted domain guarantees the feasibility of protecting data security and privacy. The detailed steps are as follows.

Step 1. The syntactic elements of *coeff_abs_level_remaining* and *absCoeffLevel* in the P-frames are first identified by parsing the bitstream.

Step 2. According to the previous embedding rules, the hidden data can be extracted as follows.

$$\overline{w\,(i)} = \begin{cases} 0 & \text{if } \overline{coeff\_abs\_level\_remaining} \ \ \%2 = 0, \\ 1 & \text{if } \overline{coeff\_abs\_level\_remaining} \ \ \%2 = 1, \end{cases} \tag{6}$$

where $\overline{w(i)}$ d denotes the extracted hidden bit. However, according to the embedding rules, the information bit cannot be extracted in the following special cases.

1. Case 1: $m = 0$ and $\overline{coeff\_abs\_level\_remaining} \leq 3$,
2. Case 2: $m = 1$ and $\overline{absCoeffLevel} = 6$ or 7,
3. Case 3: $m = 2$ and $\overline{absCoeffLevel} = 12$ or 13,
4. Case 4: $m = 3$ and $\overline{absCoeffLevel} = 24$ or 25.

Step 3. According to the data hiding key, the original message can be further extracted by decrypting the extracted hidden bits.

Step 4. According to the encryption keys, the plaintext video containing the hidden message can be obtained by decrypting the encrypted video. Because the *absCoeffLevel* is slightly modified during the data embedding process, the visual content of the decrypted video is very similar to that of the original video. This will be proven by subsequent experimental observations.

From this situation, it can be seen that the whole process can be performed completely in the encrypted domain, which effectively prevents the leakage of the media content. However, in some other situations, users need to decrypt the video first, and then extract the hidden data from the decrypted video [16,17]. This can also be realized in our scheme. First, the encrypted bin strings can be decrypted by utilizing the cipher streams used in the encryption

TABLE 6 The set of benchmark video sequences used to evaluate the performance

Class	Resolution	Frame rate	Videos
A	352 × 288	30	Football, Bus
B	832 × 480	50–60	BasketballDrill, BQMall
C	1920 × 1080	24	Kimono, Tennis
D	2560 × 1600	30	PeopleonStreet, Traffic

process. Because the XOR operation is symmetric, the decryption process is the same as the encryption process. After decryption, the value of the *absCoeffLevel* will remain unchanged. Consequently, the hidden message can be extracted using (6).

3 | EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed scheme is implemented by applying the encryption and data embedding on the HEVC reference software version HM-12.0 [27]. Table 6 defines the set of benchmark video sequences that are used in the experimental tests. The proposed encryption/data hiding scheme and HEVC compression are simultaneously performed on all the benchmark video sequences for the low-delay mode. The test conditions for encoding all the sequences are 100 frames and an intra-period of 4.

3.1 | Scrambling effect and security analysis

On the receiver's side, the authorized users can extract the hidden data. Similar to the methods in [16,17], the hidden messages can be extracted before or after video decryption. Data extraction in the encrypted domain guarantees the feasibility of protecting data privacy or security. To demonstrate the visual protection offered by the proposed encryption scheme, the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and video quality measurement (VQM) [28] are employed to evaluate the scrambling effect. In Table 7, the results of eight video sequences without encryption and with selective encryption for the quantization parameter (QP) value of 28 are presented. The average PSNR is 12.7132 dB. Numerical results show that encrypting the sign bits of the nonzero QTCs and MVDs can also effectively scramble the visual quality. The experimental results of visual scrambling are also shown in Figures 2 and 3. The original frame of each video is shown in Figure 2, and the corresponding encryption result is shown in Figure 3. Due to space constraints, we do not list the visual results of all the frames. As a supplement, the PSNR values of all the encrypted frames are depicted in Figure 4. It can be seen that the PSNR value of each encrypted frame is different, but all the values are very low.

Generally, the portion of the video frame that contains many details and textures will have many nonzero coefficients, and will therefore be highly encrypted. Conversely, the smooth areas in the video frames are relatively weakly encrypted. The selective encryption of the syntactic element, MVD, helps to protect the motion information in the video sequence. As can be seen from Figure 3 and Table 7, for high-resolution videos (ie, PeopleOnStreet, Traffic), the scrambling effect of the proposed encryption is not very good. This is attributable to the fact that the larger coding unit (CU) will be utilized for encoding the video with high resolution. Consequently, the QTCs and MVDs available for encryption are relatively small. As solution, the encryption of the IPM and that of the sign bits of the QTCs and MVDs can be fused. However, when the IPM is encrypted, the video bit rate will be slightly affected [17]. This will be further studied in our future work.

TABLE 7 The perceptual quality of the encrypted video

	PSNR (dB)		SSIM		VQM	
Sequence	**Reconstructed**	**Encrypted**	**Reconstructed**	**Encrypted**	**Reconstructed**	**Encrypted**
Football	37.1813	13.0572	0.9434	0.3196	0.8573	9.8439
Bus	35.6082	9.0887	0.9606	0.2067	0.9743	10.4857
BasketballDrill	37.8842	11.5447	0.9342	0.3545	0.8449	12.8296
BQMall	37.5068	12.2927	0.9498	0.4214	0.8566	9.2544
Kimono	40.5499	11.8267	0.9527	0.5479	0.7684	8.3539
Tennis	39.7046	12.9569	0.9436	0.5047	0.7844	7.2785
PeopleonStreet	38.3239	15.1814	0.9498	0.6579	0.7723	7.7510
Traffic	39.0604	15.7571	0.9581	0.7266	0.8420	8.1461
Average	**38.2833**	**12.7132**	**0.9539**	**0.4674**	**0.8578**	**9.2429**

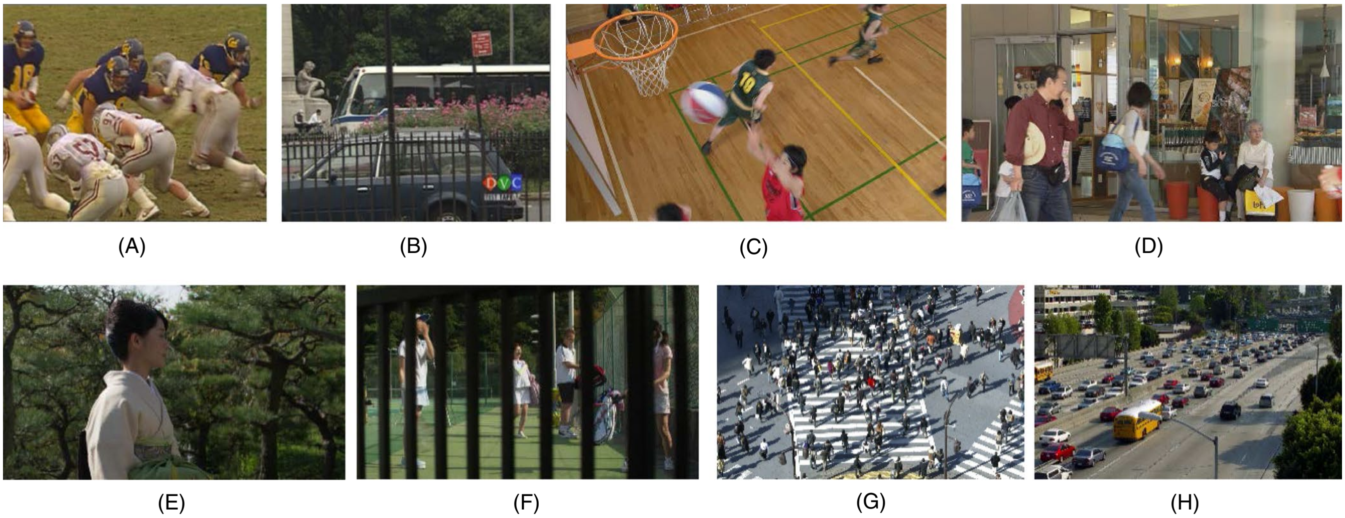


FIGURE 2 Original video frames**:** (A) *Football*: the 10th frame, (B) *Bus*: the 10th frame, (C) *BasketballDrill*: the 10th frame, (D) *BQMall*: the 10th frame, (E) *Kimono*: the 10th frame, (F) *Tennis:* the 10th frame, (G) *PeopleonStreet*: the 10th frame, and (H) *Traffic*: the 10th frame

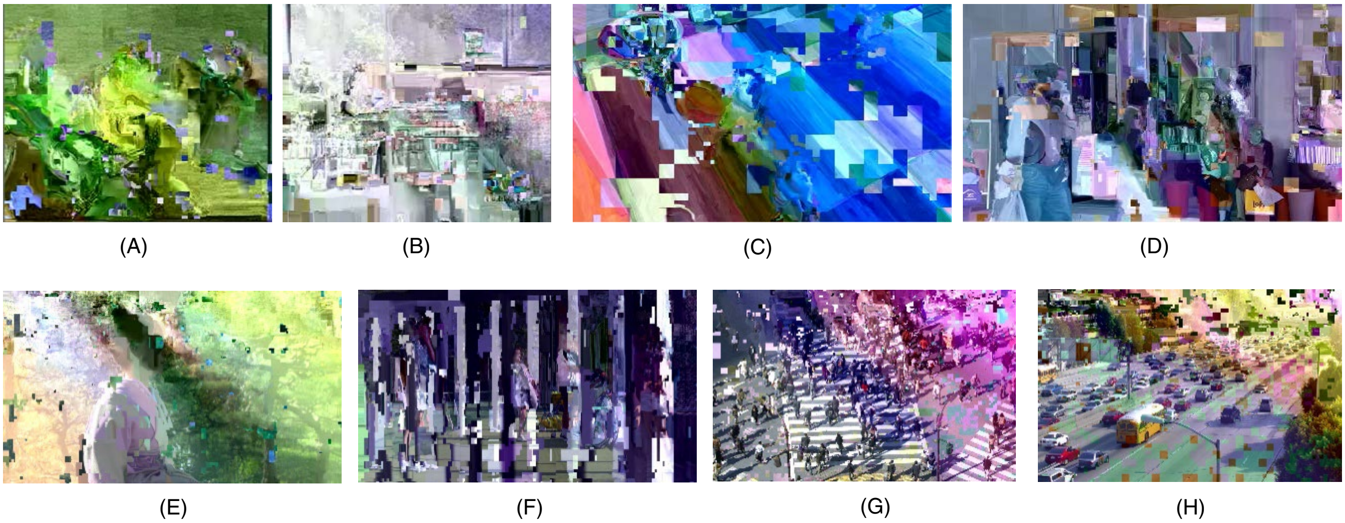
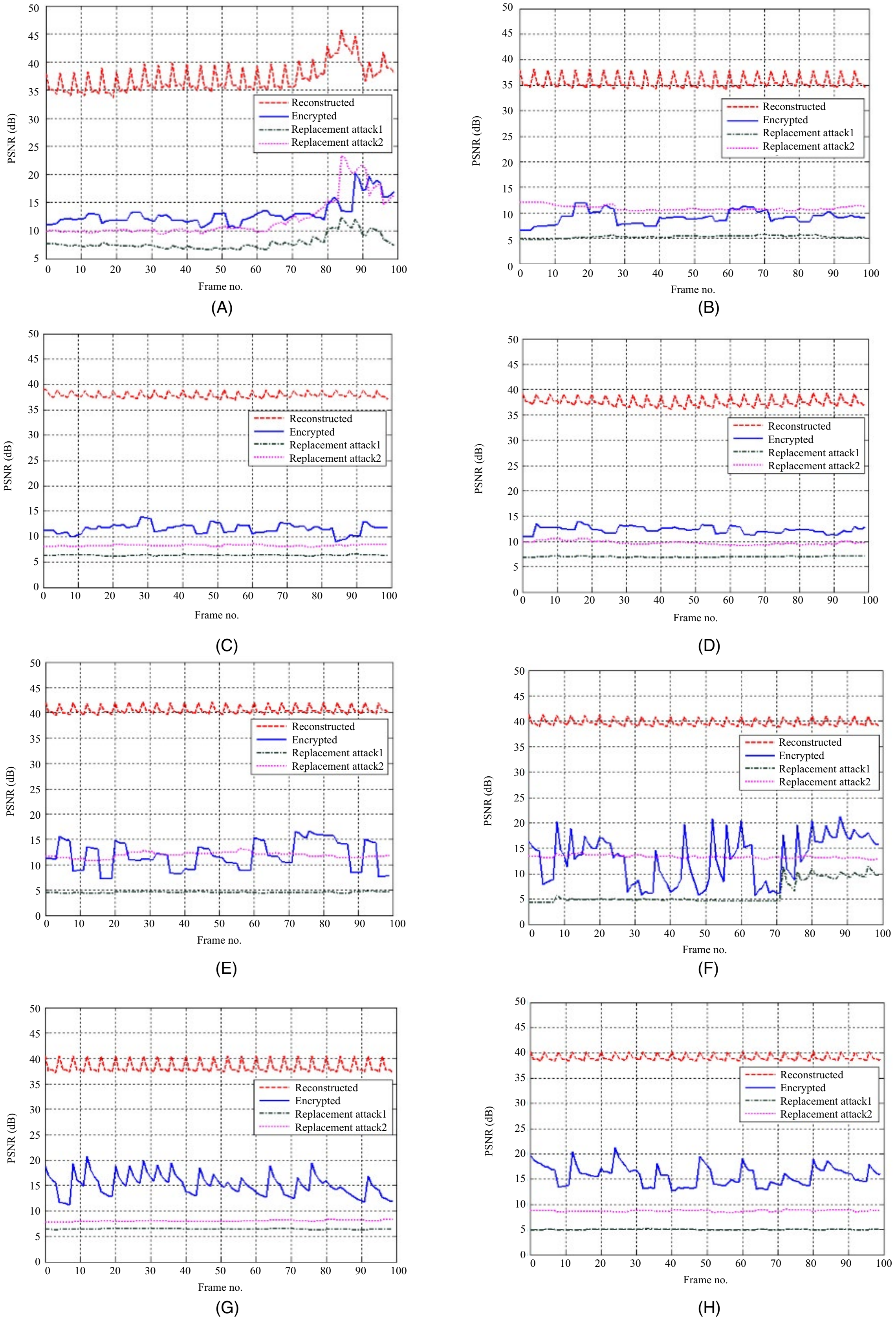


FIGURE 3 The corresponding encrypted video frames (A) *Football* (12.0638 dB), (B) *Bus* (7.6619 dB), (C) *BasketballDrill* (10.3501 dB), (D) *BQMall* (12.7485 dB), (E) *Kimono* (8.8344 dB), (F) *Tennis* (13.5962 dB), (G) *PeopleonStreet* (15.5354 dB), and (H) *Traffic* (13.6834 dB)

From the perspective of cryptography, the security of the proposed video encryption scheme depends on the adopted stream cipher. In our experiments, the pseudorandom number generator (PRNG) is utilized to generate the bitstream, and its security has been confirmed. One of the common attacks in video encryption is the replacement attack. Therefore, the robustness of the proposed algorithm against the replacement attack is further verified. Here, the replacement attack is performed by setting all the cipherable bits to "0" (marked as "Replacement Attack 1") or "1" (marked as "Replacement Attack 2"). The PSNR values of each frame following a replacement attack are also presented in Figure 4. As can be seen, after the replacement attack, the PSNR values remain very small, which means that the video content information cannot be revealed. Therefore, this proves that the proposed encryption scheme is robust against replacement attacks.

3.2 | Visual quality of decrypted video

As described in Section 2.3 the authorized user has to decrypt the encrypted videos containing hidden data in some



(A)

(B)

(C)

(D)

(E)

(F)

(G)

(H)

FIGURE 4 *PSNR* values of all encrypted frames (A) *Football*, (B) *Bus,* (C) *BasketballDrill,* (D) *BQMall,* (E) *Kimono,* (F) *Tennis,* (G) *PeopleOnStreet*, and (H) *Traffic*

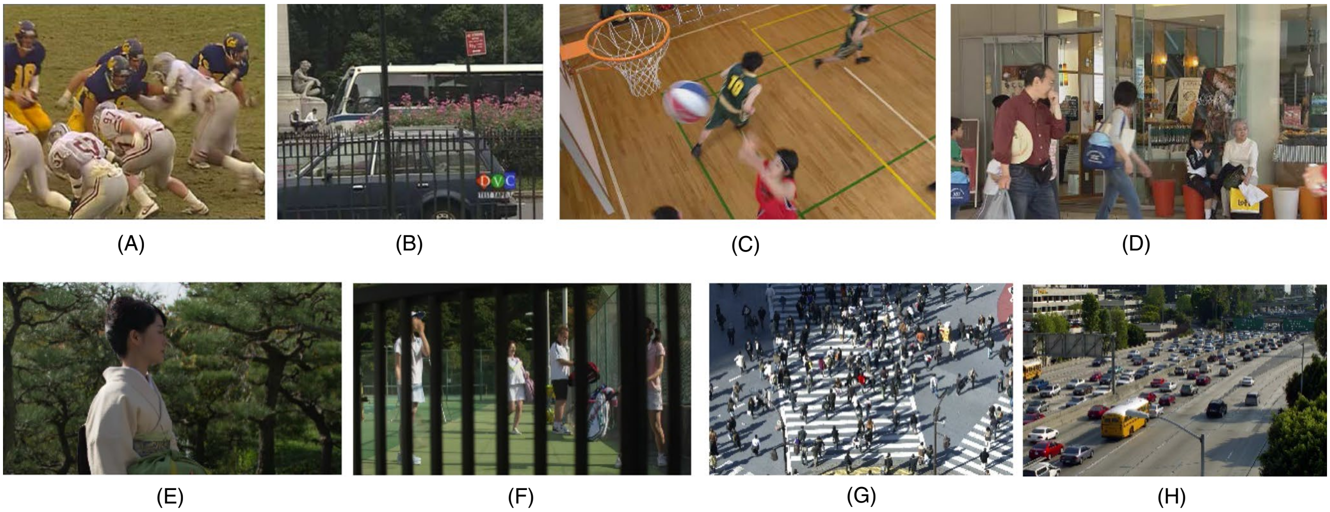


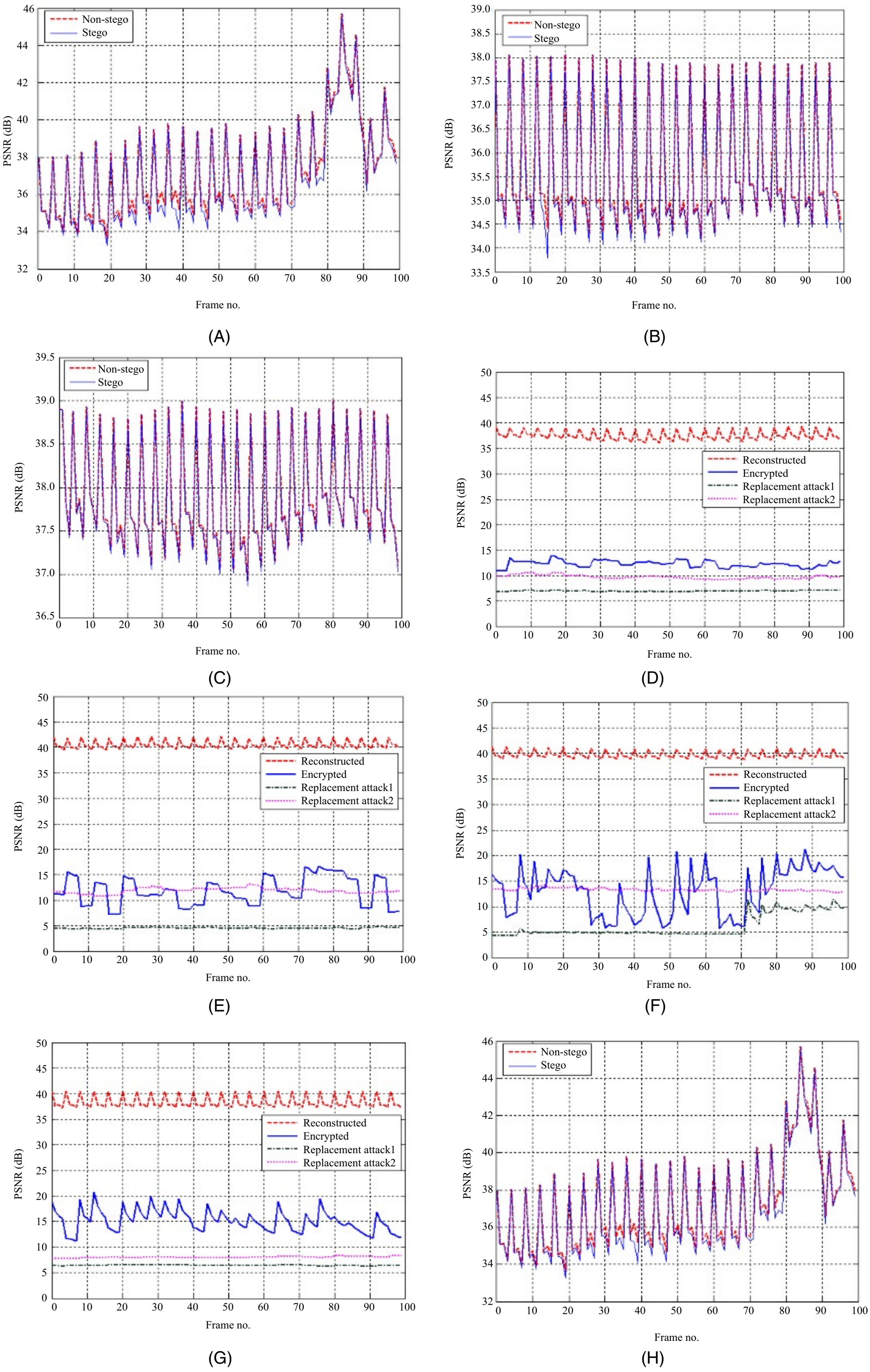
FIGURE 5 The corresponding decrypted video frames containing the hidden data (A) *Football* (34.5586 dB), (B) *Bus* (35.0419 dB), (C) *BasketballDrill* (37.6890 dB), (D) *BQMall* (37.6404 dB), (E) *Kimono* (40.0834 dB), (F) *Tennis* (39.5435 dB), (G) *PeopleonStreet* (37.7894 dB), and (H) *Traffic* (38.7298 dB)

TABLE 8 Test results of the proposed scheme

		Maximum capacity		PSNR (dB)		SSIM
Sequence	**QP**	**bits**	**kbits/s**	**Non-marked**	**Marked**	**Non-marked**
Football	28	16 902	5.0706	37.1813	36.9409	0.9434
	32	5222	1.5666	34.2510	34.1706	0.9004
Bus	28	7395	2.2185	35.6082	35.5396	0.9606
	32	1334	0.4002	32.8120	32.7967	0.9328
BasketballDrill	28	5196	2.5980	37.8842	37.8613	0.9342
	32	1595	0.7975	35.6245	35.6150	0.9013
BQMall	28	5962	3.5772	37.5068	37.4853	0.9498
	32	1722	1.0332	35.2199	35.2113	0.9272
Kimono	28	85 567	20.5361	40.5499	40.4709	0.9527

scenarios. Therefore, the decrypted videos containing the hidden data should be easily decoded with acceptable perceptual quality using a standard decoder. According to (5), because the largest modification of *absCoeffLevel* is 1 during the data embedding process, the artifacts introduced will not be perceptible. To verify this, a series of tests have also been performed. Some original video frames and their corresponding decrypted versions containing the hidden data are shown in Figures 2 and 5, respectively. It can be seen that the perceived quality of the decrypted video frame is very good, and does not appear to be different from the original video frames. Other video frames are similar in terms of visual quality. Due to space constraints, the visual results of other frames will not be listed here. Therefore, from our subjective observations, it can be concluded that marked content cannot be visually distinguished from unmarked content.

Because HEVC is lossy compression, to better demonstrate the impact of data embedding on video quality, it is necessary to test the visual quality of unmarked video streams. Usually, a video sequence obtained by decompressing an unmarked video stream (ie, reconstructed video) is used as a target sequence, and an original uncompressed video sequence is used as a reference video sequence. Similarly, to test the visual quality of marked video streams, a video sequence obtained by the encryption, data hiding, decryption, and decompression process is used as a target sequence. In other words, in this case, the target video contains hidden data. The comparison results are shown in Table 8. It is usually difficult to detect the degradation of visual quality caused by data hiding. Furthermore, the PSNR values of all the marked frames are presented in Figure 6. We can see that for motion videos (eg, Football), the impact of the data embedding process on video quality is relatively huge.



(A)

(B)

(C)

(D)

(E)

(F)

(G)

(H)

FIGURE 6 *PSNR* values of all marked frames: (A) *Football,* (B) *Bus,* (C) *BasketballDrill,* (D) *BQMall,* (E)*Kimono,* (F) *Tennis,* (G) *PeopleonStreet,* and (H) *Traffic*

3.3 | Embedding capacity

Table 8 shows the maximum embedding capacity of each video when the QP values are 28 and 32. It is expressed in units of bits and kilobits per second (kbits/s). The overall embedding capacity greatly depends on the video content. This is because the qualified coefficients (QCs) of each video available for data embedding varies. From Table 8, it can be observed that the sequences, Football, Kimono, and PeopleonStreet have significantly larger embedding capacity. The sequence, Football, has considerable motion, whereas the sequences, Kimono and PeopleonStreet, are highly textured. The number of qualified QTCs in the P-frames of these video sequences is relatively large, compared to the sequences, Bus, BasketballDrill, BQMall, and Traffic, that have limited motion in some frames and, are coded using *skip* blocks.

3.4 | Bit rate overhead

To further evaluate the performance, the bit rate overhead (BR_var) caused by encryption and data hiding was proposed [17]. It is represented as follows:

$$\mathrm{BR\_var} = \frac{\mathrm{BR\_em} - \mathrm{BR\_orig}}{\mathrm{BR\_orig}} \times 100\%, \qquad (7)$$

where BR_em is the bit rate after encryption and data embedding, and BR_orig is the original bit rate. In our experimental results, BR_em is equal to 0, which means that the bit rate remains unchanged after encryption and data embedding. This is because encryption and data hiding are performed by modifying a suitable *absCoeffLevel* to another *absCoeffLevel* with the same length of *bin string*. Specifically, the syntactic elements selected for encryption and data embedding are all encoded in the *bypass* mode, wherein a fixed context is used, as described in Section 2. In short, the process of encryption and data hiding does not affect the compression efficiency of the encoder.

3.5 | Discussion and comparative analysis

In previous work [13–19], some efficient algorithms for embedding data into encrypted H.264/AVC video using *code word/bin-string* substitution were proposed. In H.264/AVC, when the absolute value of the nonzero QTCs do not exceed 15, there are no suitable *bin strings* to be used for data hiding [16,17]. In the HEVC, only the first two bins of the coefficient level (*coeff_abs_level_greater1_flag* and *coeff_abs_level_greater2_flag*) are context-coded. The remaining portion of the levels (*coeff_abs_level_remaining*) is bypass coded [29]. Consequently, there are significantly more eligible *bin strings* in the HEVC that can provide larger capacity for data hiding. Taking Football as an example, when the QP value is 28, the maximum embedding capacities in [16] and [17] are 121 and 137 bits, and our method's can be as high as 16 902 bits. Correspondingly, due to information embedding, the PSNR values in [16] and [17] decrease by 0.01 and 0.06 dB, and by 0.24 dB in our proposed method. Obviously, the more information is embedded, the greater the impact on the quality of video perception. In addition, when the QP value is 20, the maximum embedding capacities in [16] and [17] are 4332 and 6345 bits, but our method's is 137 818 bits. We cannot give all the test results; it is however worth noting that the embedding capacities of the other videos also increased. The reason for choosing the methods in [16] and [17] for comparison is that these methods are based on the CABAC entropy coding, as is the method used in this study.

TABLE 9 Comparative analysis of existing schemes

Methods	Elements for encryption	Elements for data embedding	Bit rate increase	Separability	Reversibility	Coding standard
[13]	IPM, sign of MVD, sign of QTC	Amplitude of QTC	Yes	Yes	No	H.264/AVC
[14]	IPM, sign of MVD, sign of QTC	CAVLC code words	No	Yes	No	H.264/AVC
[15]	IPM, sign of MVD, sign of QTC	CAVLC code words	No	Yes	No	H.264/AVC
[16]	sign of MVD, sign of QTC	CABAC bin strings	No	Yes	No	H.264/AVC
[17]	IPM, sign of MVD, sign of QTC	CABAC bin strings	Yes	Yes	No	H.264/AVC
[18]	IPM, sign of MVD, sign of QTC	Amplitude of QTC	Yes	Yes	Yes	H.264/AVC
[19]	IPM, sign of MVD, sign of QTC	Amplitude of QTC	Yes	Yes	Yes	H.264/AVC
[20]	sign of MVD, amplitude of MVD, sign of QTC	Amplitude of QTC	Yes	Yes	Yes	HEVC
Proposed method	**sign of MVD, sign of QTC**	**Amplitude of QTC**	**No**	**Yes**	**No**	**HEVC**

The comparative analysis results are also listed in Table 9. It is evident that the bit rate and format compatibility can be completely preserved using the proposed method.

4 | CONCLUSION AND FUTURE WORK

Data hiding for encrypted videos in cloud computing and privacy-preserving applications have attracted growing scholarly attention. In this study, an effective CABAC-based scheme for directly embedding additional data in partially encrypted HEVC videos using the QTC modification is presented. Because the secret data are directly embedded in the encrypted domain and the encryption key is concealed, the scheme can preserve the confidentiality of video content. The selective encryption is designed to encrypt QTCs sign bits and MVDs sign bits, which has no effect on the HEVC video format compliance and bit rate. An efficient coefficient modification technique is designed for data embedding. In addition, the data extraction process can be performed regardless of whether the video is in the encrypted or decrypted domain. The security analysis results demonstrated that the encryption is secure and robust against the replacement attack. Furthermore, experimental results have also demonstrated that the degradation in the video quality owing to data hiding is quite negligible. Future work will focus on how to seamlessly integrate these key three functions of our scheme: encryption, data hiding, and HEVC video compression.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (61771270), Zhejiang Provincial Natural Science Foundation of China (LY17F020013, LR20F020001), and Ningbo Natural Science Foundation (2018A610054).

ORCID

Dawen Xu https://orcid.org/0000-0002-9619-8407

REFERENCES

1. M. Usman, M. A. Jan, and X. J. He, *Cryptography-based secure data storage and sharing using HEVC and public clouds*, Inf. Sci. **387** (2017), 90–102.
2. B. Zhao, W. D. Kou, and H. Li, *Effective watermarking scheme in the encrypted domain for buyer–seller watermarking protocol*, Inf. Sci. **180** (2010), 4672–4684.
3. J. T. Guo, P. J. Zheng, and J. W. Huang, *Secure watermarking scheme against watermark attacks in the encrypted domain*, J. Vis. Commun. Image Represent. **30** (2015), 125–135.
4. A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, *Robust watermarking of compressed and encrypted JPEG2000 images*, IEEE Trnas. Multimedia **14** (2012), no. 3, 703–716.
5. H. Liu et al., *Robust and hierarchical watermarking of encrypted images based on compressive sensing*, Signal Process. Image Commun. **45** (2016), 41–51.
6. S. J. Xiang and J. Y. He, *Database authentication watermarking scheme in encrypted domain*, IET Inf. Secur. **12** (2018), no. 1, 42–51.
7. R. M. Rad, K. S. Wong, and J. M. Guo, *A unified data embedding and scrambling method*, IEEE Trans. Image Process **23** (2014), no. 4, 1463–1475.
8. X. P. Zhang, *Separable reversible data hiding in encrypted image*, IEEE Trans. Inf. Forensics Secur. **7** (2012), no. 2, 826–832.
9. F. J. Huang, J. W. Huang, and Y. Q. Shi, *New framework for reversible data hiding in encrypted domain*, IEEE Trans. Inf. Forensics Secur. **11** (2016), no. 12, 2777–2789.
10. X. C. Cao et al., *High capacity reversible data hiding in encrypted images by patch-level sparse representation*, IEEE Trans. Cybernetics **46** (2016), no. 5, 1132–1143.
11. Z. X. Qian and X. P. Zhang, *Reversible data hiding in encrypted images with distributed source encoding*, IEEE Trans. Circ. Syst. Vid. **26** (2016), no. 4, 636–646.
12. D. W. Xu and R. D. Wang, *Separable and error-free reversible data hiding in encrypted images*, Signal Process. **123** (2016), 9–21.
13. S. G. Lian, Z. X. Liu, and Z. Ren, *Commutative encryption and watermarking in video compression*, IEEE Trans. Circ. Syst. Vid. **17** (2007), no. 6, 774–778.
14. D. W. Xu, R. D. Wang, and Y. Q. Shi, *Data hiding in encrypted H.264/AVC video streams by codeword substitution*, IEEE Trans. Inf. Forensics Secur. **9** (2014), no. 4, 596–606.
15. D. W. Xu, R. D. Wang, and Y. Q. Shi, *An improved scheme for data hiding in encrypted H.264/AVC videos*, J. Vis. Commun. Image Represent. **36** (2016), 229–242.
16. D. W. Xu and R. D. Wang, *Context adaptive binary arithmetic coding-based data hiding in partially encrypted H.264/AVC videos*, J. Electronic Imaging. **24** (2015), no. 3, 033028:1–13.
17. D. W. Xu, R. D. Wang, and Y. N. Zhu, *Tunable data hiding in partially encrypted H.264/AVC videos*, J. Vis. Commun. Image Represent. **45** (2017), 34–45.
18. D. W. Xu and R. D. Wang, *Efficient reversible data hiding in encrypted H.264/AVC videos*, J. Electronic Imaging. **23** (2014), no. 5, 053022:1–14.
19. Y. Z. Yao, W. M. Zhang, and N. H. Yu, *Inter-frame distortion drift analysis for reversible data hiding in encrypted H.264AVC Video bitstreams*, Signal Process. **128** (2016), 531–545.
20. M. Long, F. Peng, and H. Y. Li, *Separable reversible data hiding and encryption for HEVC Video*, J. Real-Time Image Process. **14** (2018), no. 1, 171–182.
21. D. Marpe, H. Schwarz, and T. Wiegand, *Context-based adaptive binary arithmetic coding in the H.264/AVC video, compression standard*, IEEE Trans. Circ. Syst. Vid. **13** (2003), no. 7, 620–636.
22. 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. Circ. Syst. Vid. **21** (2011), no. 5, 565–576.
23. Y. S. Wang, M. O. Neill, and F. Kurugollu, *A tunable encryption scheme and analysis of fast selective encryption for CAVLC and*

CABAC in H.264/AVC, IEEE Trans. Circ. Syst. Vid. **23** (2013), no. 9, 1476–1490.

24. B. Boyadjis et al., *Extended selective encryption of H.264/AVC (CABAC) and HEVC encoded video streams*, IEEE Trans. Circ. Syst. Vid. **27** (2017), no. 4, 892–906.
25. A. Sallam, O. Faragallah, and E. EL-Rabaie, *HEVC selective encryption using RC6 block cipher technique*, IEEE Trans. Multimedia. **20** (2018), no. 7, 1636–1644.
26. J. Sole et al., *Transform coefficient coding in HEVC*, IEEE Trans. Circ. Syst. Vid. **22** (2012), no. 12, 1765–1777.
27. HEVC Reference Software HM 12.0 [Online]. Available https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags [last accessed Oct. 2018].
28. 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. **7** (2012), no. 4, 205–214.
29. V. Sze and M. Budagavi, *High throughput CABAC entropy coding in HEVC*, IEEE Trans. Circ. Syst. Vid. **22** (2012), no. 12, 1778–1791.

AUTHOR BIOGRAPHY

Dawen Xu received his MS in Communication and Information System from Ningbo University, China, in 2005. He obtained a PhD degree in Computer Applied Technology from Tongji University, China, in 2011. He is a Professor at the School of Electronics and Information Engineering, Ningbo University of Technology, China. His research interests include digital watermarking and information hiding, steganalysis, and signal processing in the encrypted domain. He has served as a technical paper reviewer for IEEE conferences, journal & magazines.