International Journal of Modern Computer Science (IJMCS) ISSN: 2320-7868 (Online)
Volume 5, Issue 4, August, 2017

Secret Data Hiding in H.264/AVC Compressed Videos

S.Saraswathi,
Department of IT,
New Prince ShriBhavani College of Engg & Techn.
saraswathi1006@gmail.com

K.Blessing Christiana,
Assistant Professor, Department of IT
New Prince Shri Bhavani College of Engg & Techn.
kblesssingchristiana@gmail.com

Abstract— Digital video 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 this paper, a scheme of encrypted data hiding directly in the encrypted version of H.264/AVC video stream is proposed, which includes the following three parts, i.e., H.264/AVC video encryption, data embedding, and data extraction. Information hiding methods in H.264/AVC compressed video is illustrated by using various data representation schemes such as bit plane replacement, spread spectrum, histogram manipulation, and matrix encoding. Furthermore, video file size is strictly preserved even after encryption and data embedding. Experimental results have demonstrated the feasibility and efficiency of the proposed scheme.

Keywords: Data Hiding, Steganalysis, Video compression, Video encryption, Watermarking

I. INTRODUCTION

For higher efficiency in video coding, H.264 (H.264/advance video coding) is proposed by the Video Coding Experts Group [1] and it has become one of the most commonly practiced video compression formats since 2003. The design of H.264 provides an enhanced compression performance on video representation for various purposes, including video telephony, storage etc. broadcast, and streaming applications. H.264 achieves a significant improvement in rate distortion trade off by offering high video quality for relatively low bit rate as compared to the previous generations of video compression standards. As a result, various digital video technologies lay on the H.264 compression framework, such as Blu-ray video disc, video streaming (e.g., YouTube, Daily motion), surveillance camera, handy video recorder, etc. A similar trend is expected for the recently finalized H.265 video compression standard [2].

Currently, with the existence of broadband Internet service and ubiquitous network coverage through a cellular data plan, video can be conveniently downloaded and broadcast through social networking services such as YouTube, Facebook, and Twitter. Therefore, there are various needs to manage and/or protect the vast number of videos including [3]: 1) tracking illegal distribution of copyrighted video to securebusiness revenue; 2) hyperlinking related contents while ensuring the hyperlink information always stays intact with the video to enhance user experiences; and 3) monitoring video
broadcasts and Internet distributions to generate reports
regarding when, where, and how many times a video has been aired/streamed. It refers to the process of inserting information into a host to achieve certain features or to serve specific purposes.
The components of information hiding are shown in Fig.1.

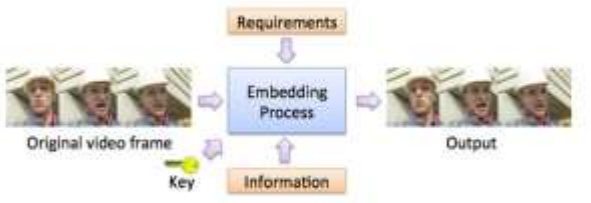


Fig 1: General frame work of information hiding

Therefore, it becomes highly desirable to develop data hiding algorithms that work entirely on encoded bit stream in the encrypted domain However, there are some significant challenges for data hiding directly in compressed and encrypted bit stream. The first challenge is to determine where and how the bit stream can be modified so that the encrypted bit stream with hidden data is still a compliant compressed bit stream. 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.

In Section II, we describe the proposed scheme, which includes three parts, i.e., H.264/AVC video encryption, data embedding and data extraction.

II. II. PROPOSED SYSTEM

Information hiding methods designed specifically for compressed video are used to illustrate possible hiding venues within the H.264 coding structure for information hiding. We considered H.264 (instead of the latest compression standard, i.e., H.265) because of its rich literatures in various applications. There are many advantages too.It is simple and it is fully compliant with the

H.264/AVC syntax using the baseline or the extended AVC profiles Message hiding works for both coded and skipped macro blocks. The proposed solution also works independent of picture type being I (intra), P (predicted) or B (bidirectional predicted).

Information hiding methods designed specifically for compressed video, illustrate possible hiding venues within the H.264 coding structure for information hiding, and review their applications. We considered H.264 (instead of the latest compression standard, i.e., H.265) because of its rich literatures in various applications. Here, we emphasize on the techniques that manipulate the

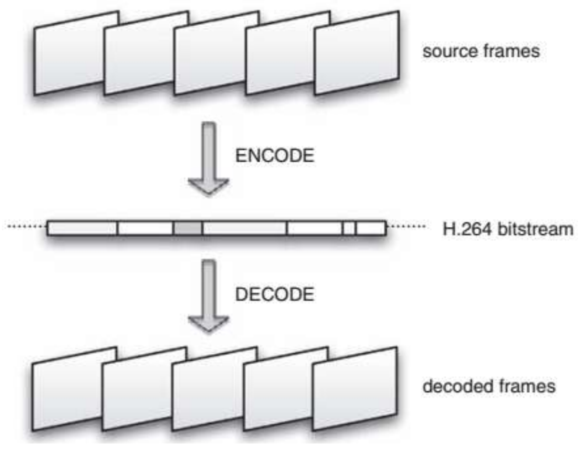


Fig 2: Video coding: source frames, encoded bit stream, decoded frames.

underlying coding structure of H.264 to realize data embedding and how each of the techniques affects the payload (i.e., the number of bits that can be inserted into the host video), bit stream size overhead, video quality and computational complexity. Nevertheless, at times, information hiding methods designed for image are also reviewed since they can be readily applied to compressed video.

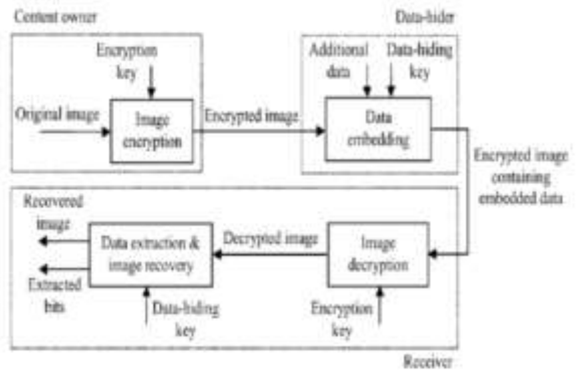
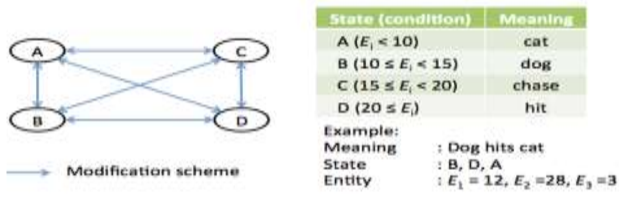


Fig 3: Overall Process

III. DATA REPRESENTATION SCHEMES

Conceptually, an information hiding method can be illustrated by the relationship among state, entity, and the meaning of each state. Here, meaning refers to part (e.g., the 16th 4-bit segment) of the information and it is represented (i.e., encoded) by a particular state of the entity in which case an entity can be a pixel, coefficient, coding mode, etc. The information determines the state in which the entity should be in during the embedding process. For the example shown in Fig.4, the message "dog hits cat" can be represented by three coefficient values, namely, E_1 =12, E_2 = 28 and E_3 =3 in the chrominance Ch.



State (condition)	Meaning
A ($E_i < 10$)	cat
B ($10 \le E_i < 15$)	dog
C ($15 \le E_i < 20$)	chase
D ($20 \le E_i$)	hit

Example:

Meaning	: Dog hits cat
State	: B, D, A
Entity	: $E_1 = 12$, $E_2 = 28$, $E_3 = 3$

Fig 4: State of the entity and its meaning for information hiding

Next, it is given the details of the commonly considered data representation schemes for information hiding, focusing on those that are applicable to video domain under the H.264 compression standard. Fig.5 shows the classification of data representation schemes in H.264 and the following subsections describe them in greater detail. Here, the dotted enclosing rectangle emphasizes that matrix encoding is applied on one or more of the data representation schemes (i.e., rectangles in solid line) in Fig. 5.

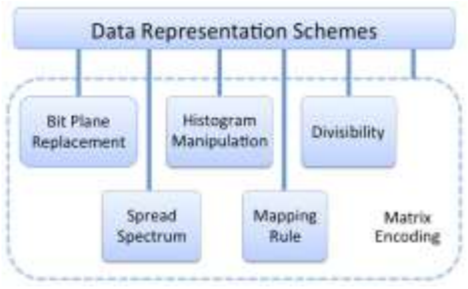


Fig 5: Classification of data representation schemes for information hiding in H.264 compressed video.

A. Bit plane replacement:

The general idea of bit plane replacement is to embed information in a particular bit plane (i.e., location) agreed upon by both the sender and receiver. In this approach, one bit can be inserted into a digital host without causing significant perceptual impact on the host content. This technique is commonly applied to entities such as raw pixel value, audio sample, and motion vector information. It achieves high payload, low distortion, and its implementation is relatively straightforward.

B. Histogram Manipulation

Fig. 10 shows the histogram of pixels in a compressed still image (equivalent to one frame in video). The pixels values are usually associated with another value to embed information.

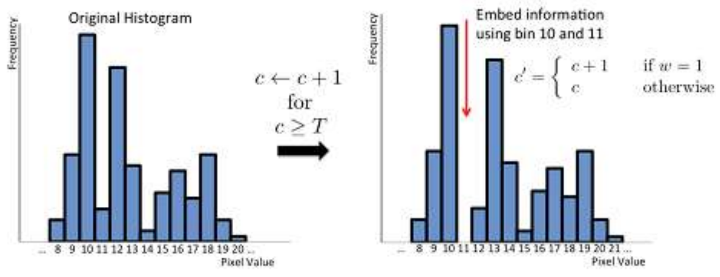
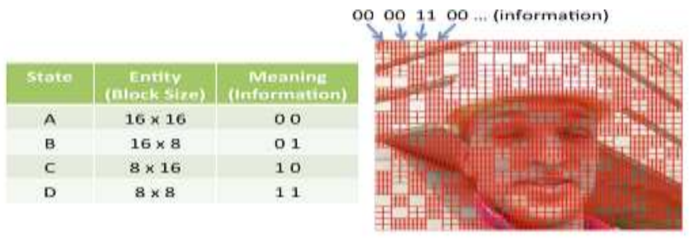


Fig 6: Hiding information using histogram [3]

In particular, Niet et al. [18] utilized the zero and peak points of the histogram and slightly modified the (grayscale) pixel values to embed information. The original histogram is preprocessed to zero and the Tth bin (i.e., the bin next to the peak point) for data embedding purposes by increasing the value c to c+ 1 for all c≥T, where T is a threshold value. Niet al. achieved reversible data embedding with high payload while maintaining high perceptual quality (i.e., >48 dB). Here, reversibility refers to the ability to perfectly reconstruct the original host.

C. Mapping Rules

Mapping rules rely on a set of code words for information embedding and extraction purposes. It depends on the structure or syntax of the content and requires a common mapping rule agreed by the sender and receiver. Basically code words are associated with predefined meaning (e.g., “00,” “01,” “10,” or “11”) and they are chosen based on the information to be embedded. Fig.11 illustrates a possible mapping rule by utilizing the macro block size to represent information.



State	Entity (Block Size)	Meaning (Information)
A	16 x 16	0 0
B	16 x 8	0 1
C	8 x 16	1 0
D	8 x 8	1 1

00 00 11 00 ... (information)

Fig 7: Mapping tools for macro blocks size to embed information [3]

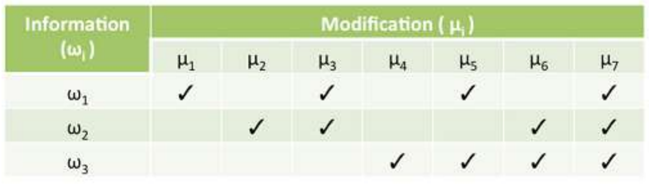
D. Divisibility

The divisibility of a value by a specific divisor can be exploited as an essential property for reversible information hiding . For example, in Wong et al. scaled the magnitude of each coefficient in the macro block by a prime number when “1” is to be embedded, or leave them as they are (i.e., no change) to embed “0”. In order for this method to be viable, divisibility of all coefficients by the chosen prime number needs to be checked to avoid error during extraction. A more sophisticated method considers a pair of neighboring pixels x and y, and the value n [22]. These values are then transformed to obey a simple equation as

$Y_i + nX_i + 1 \equiv 0 \bmod (n+1)$

F. Matrix Encoding (ME)

ME is a general principle that can be applied on top of the aforementioned data representation schemes to improve their embedding efficiencies, i.e., the number of modifications per embedded bit. Fig. 12 shows an example of the $(1,k,2^k-1)=(1,3,7)$ matrix encoding scheme and the dependencies of the information μi on the entities ω_i. Here, at most one modification is required to embed k= 3 bits using $2^k-1= 2^3-1 = 7$ entities. To embed information, μ_i is compared against the parity (denoted as pi) of the selected μi’s. The scheme then decides which μ_i to modify when necessary. For example, if $p_1 \neq \omega_1$, $p_2 \neq \omega_2$ but $p_3 = \omega_3$, then μ_3 will be modified based on the dependency stipulated in Fig. 12.



Information (ω_i)	Modification (μ_i)						
	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7
ω_1	✓		✓		✓		✓
ω_2		✓	✓			✓	✓
ω_3				✓	✓	✓	✓

Fig. 8: Example of a (1,3,7) matrix encoding

IV. MODULES

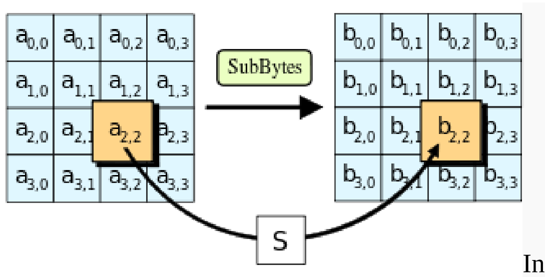
A. AES Encryption of Data:

AES (acronym of Advanced Encryption Standard) is a symmetric encryption algorithm. The algorithm was developed by two elgian cryptographers Joan Daemen and Vincent Rijmen. AES was designed to be efficient in both hardware and software, and supports a block length of 128 bits and key lengths of 128, 192, and 256 bits. To encrypt a text in the text area, set the key of the encryption then push the Encrypt button. The result of the encryption will appear in base64 encoded to prevent character encoding problems.
High-level description of the algorithm
Key Expansions—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

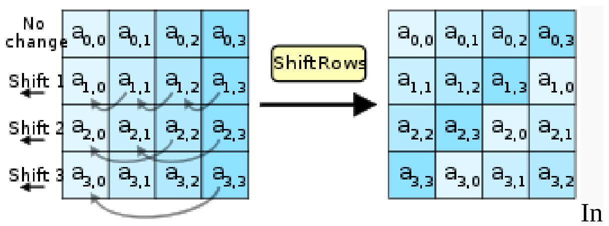
Initial Round
AddRoundKey—each byte of the state is combined with a block of the round key using bitwise xor.

1. Rounds
2. Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
3. Shift Rows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

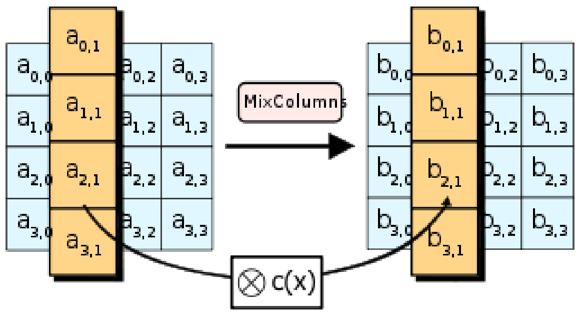
4. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
    1. AddRoundKey
5. Final Round (no MixColumns)
    1. SubBytes
    2. ShiftRows
    3. AddRoundKey.



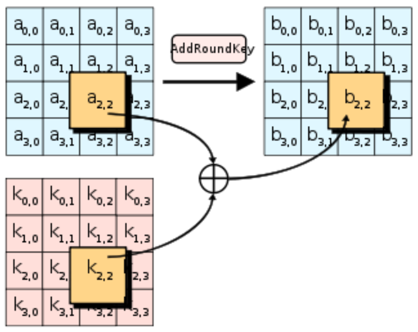
In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, *S*; $b_{ij} = S(a_{ij})$.



In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.



In the Mix Columns step, each column of the state is multiplied with a fixed polynomial *c(x)*.

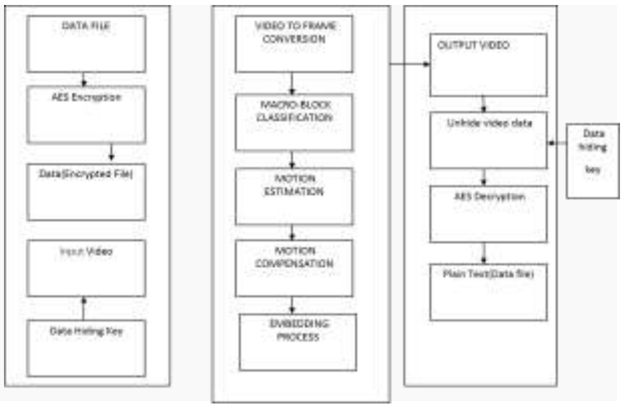


In the AddRoundKey step, each byte of the state is combined with a byte of the round sub key using the XOR operation (⊕).

Advanced Encryption Standard not only assures security but also improves the performance in a variety of settings such as smartcards, hardware implementations etc.

- AES is federal information processing standard and there are currently no known non-brute-force direct attacks against AES.
- AES is strong enough to be certified for use by the US government for top secret information.

The overall process of data hiding is depicted in the following fig,



B. Video To Frame Conversion:

The video which is given as the input file is splitted into separate frames using the slicing concept.In general, a coded picture is divided into one or more slices. Slices are self-contained and can be decoded and displayed independently of other slices. Hence, intraprediction of DCT coefficients and coding parameters of a macro block is restricted to previous macro blocks within the same slice. This feature is important to suppress error propagation within a picture due to the nature of variable length coding. In regular encoding, when FMO is not used, slices contain a sequence of macro blocks in raster scan order. However, FMO allows the encoder to create what is known as slice groups. Each slice group contains one or more slices and macro blocks can be assigned in any order to these slices. The assignment of macro blocks to different groups is signaled by a syntax structure called the “slice group id”.

C. Frame Selection

The frame selection is done based on the following two processes called motion estimation and motion compensation. Motion estimation is the process which computes the mere changes between the successive frames. The changes between the frames are estimated using the Block Matching Algorithm. This motion estimation can be done in two ways (i) Global Motion Estimation and (ii) a specific parts of an image can also be estimated. Motion vectors are used for storing the amount of motion. Motion compensation is a technique which is used to generate a frame from the estimated difference which is stored in the motion vector.

A Block Matching Algorithm (BMA) is an algorithm which identifies the matching blocks in a sequence of video frames. The main objective of this algorithm is to estimate the motion of the frames. MA identifies the temporal redundancy in a video sequence which can be used to increase the effectiveness of interframe video compression. The purpose of a block matching algorithm is to find a matching block from a frame i in some other frame j, which may appear before or after i. Block matching algorithms make use of an computation metric to determine whether a given block in frame j matches the search block in frame i[5] The Mean squared error of an estimator is the cost function which computes the average of the squares of the errors. This MSE value estimates the corresponding difference between the frames. The cost function is Mean Squared Error (MSE) given

$$MSE = \frac{1}{S^2}\sum_{i=0}^{S-1}\sum_{j=0}^{S-1}\left(P_{ij} - Q_{ij}\right)^2 \text{---------(1)}$$

by the equation

Where S is the side of the macro block, Pij and Qij are the pixels being compared in present macro block and reference macro block, respectively.

Peak-Signal-To-Noise-Ratio (PSNR) represents the motion compensated frame build using motion vectors and macro blocks from the reference frame.

$$PSNR = 10Log_{10}\left[\frac{(\text{Peak to peak value of original data})^2}{MSE}\right] \text{--------(2)}$$

Thus Motion compensation conveys the information to generate the successive frame from the reference frame and helps in identifying the place to conceal the information.

Embedding capacity per frame (EC)

EC = Data Embedding capacity per macro block × Number of macro blocks

After finding embedding capacity per frame, total video capacity can be easily obtained by multiplying number of frame.

Total video capacity (TVC)

TVC= Embedding per frame × number of frame.

D. Embedding Process :

After compression all the frames are again combined into a video. The video with concealed information appears as the original video with slight distortion. Finally the concealed information along with the video is transferred to the authorized user.

Algorithm for Data Embedding:

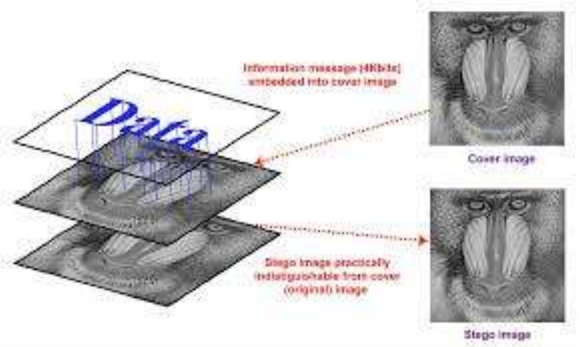
1. Partition the entire image into pairs of pixels (for instance, on rows, on columns).
2. Get the image data and txt data.
3. Convert txt data into a linear binary vector.
4. Convert the image data array into a single column vector.
5. If pixel value is odd make it even
6. Else leave it as such
7. Modifies the low bits of the values of the macro blocks / pixel image data in such a way that the modified bits from the hidden data.
8. Reshape the pixel image data into a block array.

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 code words of *IPMs*, *MVDs,*

Step3: The decryption process is identical to the encryption process, since XOR operation is symmetric. The encrypted can be decrypted by performing with generated encryption streams and AES decryption process, which renders the original plaintext. Since the encryption streams depend on the encryption keys, the decryption is possible only for the authorized users. After generating the video with hidden data, the content owner can further extract the hidden information.


Data
Information message (4Kbits) embedded into cover image
Cover image
Stego image practically indistinguishable from cover (original) image
Stego image

E. Data Retrieval and Decryption

V. CONCLUSION

Data hiding in encrypted media is a new topic that has started to draw attention because of the privacy-preserving requirements from cloud data management. In this paper, an algorithm to embed additional data in encrypted H.264/AVC bit stream is presented, which consists of video encryption, data embedding and data extraction phases. 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] W. J. Lu, A. Varna, and M. Wu, “Secure video processing: Problems andchallenges,” in Proc. IEEE Int. Conf. Acoust.,

Speech, Signal Processing, Prague, Czech Republic, May 2011, pp. 5856–5859.

[2] B. Zhao, W. D. Kou, and H. Li, "Effective watermarking scheme inthe encrypted domain for buyer-seller watermarking protocol," Inf. Sci.,vol. 180, no. 23, pp. 4672–4684, 2010.

[3] P. J. Zheng and J. W. Huang, "Walsh-Hadamard transform in the homomorphicencrypted domain and its application in image watermarking,"in Proc. 14th Inf. Hiding Conf., Berkeley, CA, USA, 2012, pp. 1–15.

[4] W. Puech, M. Chaumont, and O.Strauss, "A reversible datahiding method for encrypted images," Proc. SPIE, vol. 6819,pp. 68191E-1–68191E-9, Jan. 2008.

[5] X. P. Zhang, "Reversible data hiding in encrypted image," IEEE SignalProcess. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.

[6] W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible datahiding in encrypted images using side match," IEEE Signal Process.Lett., vol. 19, no. 4, pp. 199–202, Apr. 2012.

[7] X. P. Zhang, "Separable reversible data hiding in encryptedimage," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2,pp. 826–832, Apr. 2012.

[8] K. D. Ma, W. M. Zhang, X. F. Zhao, N. Yu, and F. Li, "Reversible datahiding in encrypted images by reserving room before encryption," IEEETrans. Inf. Forensics Security, vol. 8, no. 3, pp. 553–562, Mar. 2013.

[9] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robustwatermarking of compressed and encrypted JPEG2000 images," IEEETrans. Multimedia, vol. 14, no. 3, pp. 703–716, Jun. 2012.