ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Implementation of Reed Solomon Algorithm over Data Hiding in Video Streams

Shana Jebin P, Shyni K

P.G Scholar, Dept. of Computer Science and Engineering, Cochin College of Engineering, Kerala, India

Assistant Professor, Dept. of Computer Science and Engineering, Cochin College of Engineering, Kerala, India

ABSTRACT: In order to maintain the security and privacy of digital video, they are needed to be stored and processed in an encrypted format. For the purpose of content notation or tampering detection, it is necessary to perform data hiding in these encrypted videos. In this way, data hiding in encrypted domain without decryption preserves the confidentiality of the content. In addition, it is more efficient without decryption followed by data hiding and re-encryption. Here, a novel scheme of data hiding directly in the encrypted version of H.264/AVC video stream is proposed, which includes the following three parts, i.e., video encryption, data embedding, and data extraction. By analyzing the property of H.264/AVC codec, the codewords of intraprediction modes, the codewords of motion vector differences, and the codewords of residual coefficients are encrypted with stream ciphers. Then, a data hider may embed additional data in the encrypted domain by using codeword substitution technique, without knowing the original video content. In order to accept different application scenarios, data extraction can be done either in the encrypted domain or in the decrypted domain. For the corrupted data we use reed solomon encoding. It makes the video error free and can be used for encoding. Even after encryption and data embedding, the video file size is strictly preserved. Various experimental results conducted show the feasibility and efficiency of the proposed video encryption scheme.

KEYWORDS: Encryption, tampering detection, data hiding, data extraction, codeword, reed solomon encoding.

I. INTRODUCTION

The major technology trend, nowadays which can provide highly efficient computation and storage solution for video data in large scale is cloud computing. These cloud devices are vulnerable to untrustworthy system administrators. The main aim is to access the video content in encryption form. The leakage of video content can be avoided by concealing the data directly in encrypted video streams, which is helpful in addressing privacy concern and security with cloud computing. The embedding of some additional information in to cloud server like video notation or authentication data into a video of encrypted version by using data hiding technique is possible. By using this hidden information the video can be managed by server and also verify its integrity without knowing original content, so that the protection of security and privacy can be done. The data concealing technology can also be applied for other applications like medical videos, surveillance videos which can be encrypted for the protection and privacy of people, by embedding the personal information in to corresponding encrypted videos for providing the capabilities of data management in encrypted domain.

Data hiding technique can be used by a cloud server to embed the additional information (e.g., video notation, or authentication data) into an encrypted version of an H.264/AVC video. With the hidden information, the server can manage the video or authenticate its integrity without knowing the original video content, and thus the security and privacy can be protected. The security and data integrity are the main concerns faced in the present computing environment. In addition to cloud computing, this technology can be used in various important applications. For example, In order to protect the privacy of the people, medical videos or surveillance videos are encrypted; a database manager may embed the personal information into encrypted videos to provide the data management capabilities in the encrypted domain. The data hiding methodology gives the integrity that the original content has not been altered. So a combination of encryption and data embedding meets out the need of today's users not only in cloud computing but also for various other methods. During the encryption of the data, Firstly, fully layered Encryption compresses and then

DOI: 10.15680/IJIRCCE.2017. 0504269

encrypts the whole content by encrypting every byte using standard tradition algorithms. But it involves heavy computation and hence the speed is slow. The next method is selective encryption which encrypts only the selective features of each video. And finally, the perceptual encryption which is used in areas likes pay-per-view video, pay TV and video on demand. This kind of encryption requires that quality of audio and visual data is only partially degraded by encryption i.e. from the encrypted multimedia data one can perceive the content in it.

II. RELATED WORK

Encryption is an effective and popular means of privacy protection. In order to securely share a secret image with other person, a content owner may encrypt the image before transmission. In some application scenarios [1], an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. For example, when medical images have been encrypted for protecting the patient privacy, a database administrator may aim to embed the personal information into the corresponding encrypted images. It may be also hopeful that the original content can be recovered without any error after decryption and retrieve of additional message at receiver side.

Traditionally, data hiding is used for secret communication. In [2], a watermarking scheme in the encrypted domain using Paillier cryptosystem is suggested based on the security requirements of buyer-seller watermarking protocols. A Walsh-Hadamard transform based image watermarking algorithm in the encrypted domain using Paillier cryptosystem is presented in [3]. However, due to the constraints of the Paillier cryptosystem, the encryption of an original image results in a high overhead in storage and computation. Several researches on reversible data hiding in encrypted images are mentioned in [4], [5], [6] recently.

As all the above mentioned works are mainly focused on image, With the increasing demands of providing video data security and privacy protection, data hiding in encrypted H.264/AVC videos will surely become popular in the coming world. Due to the difficulty faced during the underlying encryption, it is very difficult and sometimes impossible to transplant the existing data hiding algorithms to the encrypted domain. To the best of our knowledge, there has been no report on the implementation of data hiding in encrypted H.264/AVC video streams. Only few joint data-hiding and encryption approaches that focus on video have been proposed. For example, in [7], during H.264/AVC compression, the intra-prediction mode (IPM), motion vector difference (MVD) and DCT coefficients' signs are encrypted, while DCT coefficients' amplitudes are watermarked adaptively. In [8], a combined scheme of encryption and watermarking is presented, which can provide the access right as well as the authentication of video content at the same time.

Another one proposes a method of reversible data hiding method in encrypted images using DCT [9]. Reversible data hiding is a technique to embed additional message into some cover media with a reversible manner so that the original cover content can be perfectly regained after extraction of the hidden message. The data extraction can be achieved by examining the block smoothness. This letter adopts a scheme for measuring the smoothness of blocks, and uses the closest match scheme to further reduce the error rate of extracted-bits. The experimental results reveal that the proposed method offers better performance over side match method. For example, when the block size is set to 88, the error rate of the image of the proposed method is zero percentage, which is significantly lower than 0.34 percentage of side match method. Reversible data hiding in images is a method that hides data in digital images for secret communication. It is a technique to hide additional message into cover media with a reversible manner so that the original content can be perfectly restored after extraction of the hidden message. Data hiding is used for secret communication.

In some applications, the embedded carriers are further encrypted to prevent the carrier from being analyzed to reveal the presence of the embodiment. Other applications could be for when the owner of the carrier might not want the other person, including data hider, to know the content of the carrier before data hiding is actually performed, such as military images or confidential medical images. In this case, the content owner has to encrypt the content before passing to the data hider for data embodiment. The receiver side can extract the embedded message and recover the original image. Many reversible data hiding methods have been proposed recently. They embed data bits by expanding the difference of two consecutive pixels. Another one uses a lossless compression technique to create extra spaces for carry data bits. Third method shifts the bins of image histograms to leave an empty bin for data embodiment. It adopts the difference expansion and histogram shifting for data embodiment. Another method embeds data by shifting the histogram of prediction mistakes while considering the local activity of pixels to further enhance the quality of stego image.

In lossless generalized-LSB data embedding [10], present a novel lossless (reversible) data-embedding technique, which enables the exact recovery of the original host signal upon extraction of the embedded information. A generalization of the well-known least significant bit (LSB) modification is proposed as the data-embedding method, which introduces additional operating points on the capacity-distortion curve. Lossless recovery of the original is achieved by compressing portions of the signal that are susceptible to embedding distortion and transmitting these compressed descriptions as a part of the embedded payload. A prediction-based conditional coder which utilizes unchanged portions of the host signal as side-information improves the compression efficiency and thus, the lossless data-embedding capability. The watermark embedding process is composed of three steps: block selection, coefficient selection and watermark embedding. In previous system, several embedding methods have been reported, which can be used to watermark the selected coefficients amplitude. The difficulty is to make watermarking and encryption commutative. As coefficients signs are encrypted, iterated watermark embedding should not affect coefficient signs. It is a method based on quantization embedding.

III. PROPOSED ALGORITHM

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

Digital video sometimes needs to be stored and processed in an encrypted format to maintain security and privacy. For the purpose of content notation and/or tampering detection, it is necessary to perform data hiding in these encrypted videos. In this way, data hiding in encrypted domain without decryption preserves the confidentiality of the content. In addition, it is more efficient without decryption followed by data hiding and re-encryption. In this paper, a novel scheme of 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. By analyzing the property of H.264/AVC codec, the codewords of intraprediction modes, the codewords of motion vector differences, and the codewords of residual coefficients are encrypted with stream ciphers. Then, a data hider may embed additional data in the encrypted domain by using codeword substitution technique, without knowing the original video content. In order to adapt to different application scenarios, data extraction can be done either in the encrypted domain or in the decrypted domain. Furthermore, video file size is strictly preserved even after encryption and data embedding. Experimental results have demonstrated the feasibility and efficiency of the proposed scheme.

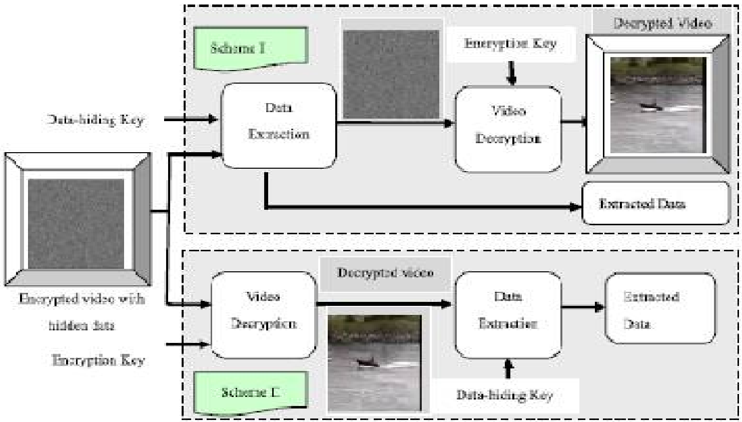

Figure 1: Data Extraction

DOI: 10.15680/IJIRCCE.2017. 0504269

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

1. Encryption of H.264/AVC Video Stream

Video encryption often requires that the scheme be time efficient to meet the requirement of real time and format compliance. It is not practical to encrypt the whole compressed video bitstream like what the traditional ciphers do because of the following two constraints, i.e., format compliance and computational cost. Alternatively, only a fraction of video data is encrypted to improve the efficiency while still achieving adequate security. The key issue is then how to select the sensitive data to encrypt. According to the analysis given, it is reasonable to encrypt both spatial information (IPM and residual data) and motion information (MVD) during H.264/AVC encoding. A H.264/AVC video encryption scheme with good performance including security, efficiency, and format compliance is proposed. By analyzing the property of H.264/AVC codec, three sensitive parts (i.e., IPMs, MVDs, and residual coefficients) are encrypted with stream ciphers. The encryption algorithm is performed not during H.264/AVC encoding but in the compressed domain. In this case, the bitstream will be modified directly. Selective encryption in the H.264/AVC compressed domain has been already presented on context adaptive variable length coding (CAVLC) and context-adaptive binary arithmetic coding (CABAC). In this paper, we have improved and enhanced the previous proposed approach by encrypting more syntax elements. We encrypt the codewords of IPMs, the codewords of MVDs, and the codewords of residual coefficients. The encrypted bitstream is still H.264/AVC compliant and can be decoded by any standard compliant H.264/AVC decoder, but the encrypted video data is treated completely different compared to plaintext video data. In fact, performing the format-compliant encryption directly on the compressed bitstream is extremely complicated as the internal states of the encoder have to be preserved, otherwise the remaining data is interpreted falsely which may easily lead to format violations.

- Intra-Prediction Mode (IPM) Encryption

  According to H.264/AVC standard, the following four types of intra coding are supported, which are denoted as Intra44, Intra1616, Intra-chroma, and I-PCM. Here, IPMs in the Intra 44 and Intra1616 blocks are chosen to encrypt. Four intra prediction modes (IPMs) are available in the Intra1616. The IPM for Intra1616 block is specified in the mb-type (macro block type) field which also specifies other parameters about this block such as coded block pattern (CBP).
- Motion Vector Difference (MVD) Encryption.

  In order to protect both texture information and motion information, not only the IPMs but also the motion vectors should be encrypted. In H.264/AVC, motion vector prediction is further performed on the motion vectors, which yields MVD. In H.264/AVC baseline profile, Exp-Golomb entropy coding is used to encode MVD. The codeword of Exp-Golomb is constructed as[M zeros] [INFO], where INFO is an M-bit field carrying information.
- Residual Data Encryption

  In order to keep high security, another type of sensitive data, i.e., the residual data in both I-frames and P-frames should be encrypted. In this section, a novel method for encrypting the residual data based on the characteristics of codeword is presented.

2. Data Embedding

Although few methods have been proposed to embed data into H.264/AVC bitstream directly, however, these methods cannot be implemented in the encrypted domain. In the encrypted bitstream of H.264/AVC, the proposed data embedding is accomplished by substituting eligible codewords of Levels. Since the sign of Levels are encrypted, data hiding should not affect the sign of Levels. Besides, the codewords substitution should satisfy the following three limitations. First, the bitstream after codeword substituting must remain syntax compliance so that it can be decoded by standard decoder. Second, to keep the bit-rate unchanged, the substituted codeword should have the same size as the original codeword. Third, data hiding does cause visual degradation but the impact should be kept to minimum. That is, the embedded data after video decryption has to be invisible to a human observer. So the value of Level corresponding to the substituted codeword should keep close to the value of Level corresponding to the original codeword. In addition, the codewords of Levels within P-frames are used for data hiding, while the codewords of Levels in I-frames are remained unchanged. Because I-frame is the first frame in a group of pictures (GOPs), the error occurred in I-frame will be propagated to subsequent P-frames.

3. Data Extraction

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

Scheme1: Encrypted Domain Extraction.

To protect privacy, a database manager (e.g., cloud server) may only get access to the data hiding key and have to manipulate data in encrypted domain. Data extraction in encrypted domain guarantees the feasibility of our scheme in this case.

- Step1: The codewords of Levels are firstly identified by parsing the encrypted bitstream.
- Step2: If the codeword belongs to code space C0, the extracted data bit is 0. If the codeword belongs to code space C1, the extracted data bit is 1.
- Step3: According to the data hiding key, the same chaotic pseudo-random sequence P that was used in the embedding process can be generated.

Scheme2: Decrypted Domain Extraction.

In scheme I, both embedding and extraction of the data are performed in encrypted domain. However in some cases, users want to decrypt the video first and extract the hidden data from the decrypted video. For example, an authorized user, which owned the encryption key, received the encrypted video with hidden data. The received video can be decrypted using the encryption key. That is, the decrypted video still includes the hidden data, which can be used to trace the source of the data. Data extraction in decrypted domain is suitable for this case.

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 bitstream is presented, which consists of video encryption, data embedding and data extraction phases. The algorithm can preserve the bit-rate exactly even after encryption and data embedding, and is simple to implement as it is directly performed in the compressed and encrypted domain, i.e., it does not require decrypting or partial decompression of the video stream thus making it ideal for real-time video applications.

IV. REED SOLOMON ENCODING

Contiguous errors in the bit stream are a common occurrence in digital communication systems, broadcasting systems and digital storage devices. Many mechanisms have devised to mitigate this problem. Forward error correction is a technique in which redundant information is added to the original message, so that some errors can be corrected at the receiver, using the added redundant information. Reed Solomon Encoder and Decoder falls in the category of forward error correction encoders and it is optimized for burst errors rather than bit errors. Reed Solomon Encoder and Decoder provide a compromise between efficiency and complexity, so that this can be easily implemented using hardware or FPGA. Reed Solomon code is based on this system. This system first discusses the Galois Field (GF) arithmetic first, and then goes into the mathematical theory behind Reed Solomon Encoder and Decoder. Here we use reed solomon encoding technique for the error detection and correction during the transmission of data.

Reed Solomon codes are non-binary, BCH, cyclic, linear block error correction codes. The major characteristics of linear block codes are block architecture, optional systematic structure, and all code words are sums of code words. It has a block length of n s ols and a message length of k symbols. If the code is systematic, then it also has an unaltered data field of k symbols independent of the associated parity check field of n-k symbols.

DOI: 10.15680/IJIRCCE.2017. 0504269

Reed Solomon Algorithm:

Input: corrupted video
Output: Actual video without error
Step 1: Give input video which is corrupted
Step 2: Set a value for the syndrome calculation block
Step 3: Check if syndrome calculation value is zero.
Step 4: If value is zero then the video has no errors else it is found to be corrupted
Step 5: In the KES block find the value for the error locator and error evaluator polynomial
Step 6: The chien search block calculates the roots of the error locator polynomial
Step 7: Finally forney block calculate the magnitude of the error symbol at each error location

Reed Solomon algorithm is used for the error detection and correction during the transmission of video through a noisy channel. It has a reed solomon encoding part and reed solomon decoding part. In Reed Solomon encoding a parity bit is added along with the data. Reed Solomon decoder is used to detect and correct the error in the data transmitted in a noisy channel. It ensures the error correction in digital communication systems and all the arithmetic operations are done by Euclidean Field.

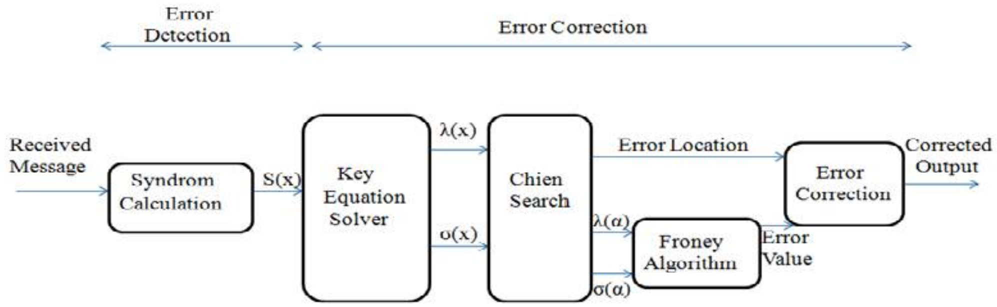

Figure 2: Reed Solomon Encoding

The proposed Reed Solomon decoder architecture consists of syndrome computation block, KES block, and error correction block as shown in Figure. The syndrome computation block and error correction block are reformulated for the eight parallel processing. To support the variable length shortened RS codes, the syndrome computation block includes permutation module and error correction block includes ROMs which store the first roots. This root value will be used to generate the error locator polynomial and error value polynomial. The root values outputted from the ROMs are increased by multiple of eight through constant multiplier and again will be used for the error locator polynomial and error value polynomial.

1. Syndrome Computation Block

The syndrome computation block calculates all the syndromes S(O <= i <= 15) by putting the roots of generator polynomial G(x) into the received codeword polynomial R(x). Equations are reformulated for eight parallel processing. Eight-parallel codeword symbols [A, B, C, D, E, F, G, H] are inputed during specific clocks that is decided by code size signal. The codeword size is variable length. If the count of codeword symbol is not multiple of eight, the codeword symbol must be permutated by adding zero-padding. The permutation is decided by code size signal and the zero symbols padded. The proposed syndrome computation block can process the syndromes during a specific clock cycles provided by codeword size information.

2. Key Equation Solver Block

Most conventional high-speed RS decoders have used an ME algorithm for KES block, because an ME algorithm can easily be implemented by fully pipelined systolic array architecture. The pipe lined degree computationless modified Euclidean (PDCME) algorithm and architecture has been used to obtain the error locator polynomial (j(x) and the error value polynomial w(x) by solving the key equation w(x) = S(x) (j(x) mod XZI. The three-stage pipelined KES block is adapted to reduce the latency, so that (j(x) and w(x) value can be outputted after 48 clock cycles.

3. Chien Search and Error Correction Block

After the KES block, the error locator polynomial j(x) and the error value polynomial w(x) are fed into the Chien search block, which calculates the roots of the error locator polynomial. The Forney algorithm block works in parallel with the Chien search block to calculate the magnitude of the error symbol at each error location. The decoder can find the error locations by checking whether =O for each j, 0 s.j S. n-l (a□1)(a255 = 0 means that r0 is corrupted by error.

V. Conclusion

The hiding of data in encrypted media is a new topic that has emerged an attention because of the privacy-preserving requirements from cloud data management. In codeword substitution based hiding, an algorithm is used to embed additional data in encrypted H.264/AVC video bit stream, which consists of video encryption, data embedding and data extraction stages. The algorithm can preserve the bit-rate exactly even after encryption and data embedding, and is simple to implement as it is directly performed in the compressed and encrypted domain, i.e., it does not require decrypting or partial decompression of the video stream thus making it ideal for real-time video applications. The data-hider can embed additional data into the encrypted bitstream using codeword substitution even he does not know the original video content. Since data hiding is completed entirely in the encrypted domain, our method can preserve the confidentiality of the content completely. With an encrypted video containing hidden data, data extraction can be carried out either in encrypted or decrypted domain, which provides two different practical applications. Another advantage is that it is fully compliant with the H.264/AVC syntax. The error detection and correction can be done using the reed solomon encoding. 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. Along with that the system provides error detection and correction in the data by using the reed solomon encoding technique.

References

1. Sarwate, D.V., and Shanbhag, N.R High-speed architectures for Reed- Solomon decoders , IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2001, 9, (5), pp. 641655
2. B. Zhao, W. D. Kou, and H. Li, “Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol,” *Inf. Sci.*, vol. 180, no. 23, pp. 4672–4684, 2010.
3. P. J. Zheng and J. W. Huang, “Walsh-Hadamard transform in the homomorphic encrypted 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 data hiding 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 Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.
6. W. Hong, T. S. Chen, and H. Y. Wu, “An improved reversible data hiding in encrypted images using side match,” *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.
7. S. G. Lian, Z. X. Liu, and Z. Ren, “Commutative encryption and watermarking in video compression,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 774–778, Jun. 2007.
8. S. W. Park and S. U. Shin, “Combined scheme of encryption and watermarking in H.264/scalable video coding (SVC),” *New Directions Intell. Interact.Multimedia*, vol. 142, no. 1, pp. 351–361, 2008.
9. X. Zhang Reversible data hiding in encrypted images, IEEE Signal Process. Lett.,vol. 18, no. 4, pp. 255258, Apr. 2011.
10. Mehmet Utku Celik,,Sharma and Ahmet Murat Tekalp,Lossless Generalized-LSB Data Embedding, IEEE Ttanscations on image processing Vol.14 , NO. 2,February 2005.
11. W. J. Lu, A. Varna, and M. Wu, “Secure video processing: Problems and challenges,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*,Prague, Czech Republic, May 2011, pp. 5856–5859.
12. Dawen Xu, Rangding Wang and Yun Q. Shi,Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution, IEEE Transactions on information forensics and security, vol. 9, no. 4, april 2014
13. S. G. Lian, Z. X. Liu and Z. Ren,Commutative encryption and watermarking in video compression, IEEE Trans. Circuits Syst. Video Technol.,vol. 17, no. 6, pp. 774778, Jun. 2007.
14. T. Wiegand, G. J. Sullivan, G. Bjontegaardand A. Luthra ,Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst.Video Technol., vol. 13, no. 7, pp. 560576, Jul. 2003.
15. Wei Liu,, Wenjun Zeng, Lina Dong and Qiuming YaoEfficient Compression of Encrypted Grayscale ImagesIEEE Ttanscations on image processing,Vol. 19, NO. 4, APRIL 2010