DATA HIDING IN ENCRYPTED H.264/AVC VIDEO STREAMS BY CODEWORD SUBSTITUTION

Mohan Kumar S. S[1] , Kusuma R[2]

[1]PG Student, [2]Assistant Professor, Dept. of CSE, Kalpataru Institute of Technology, (India)

ABSTRACT

Digital video sometimes needs to be stored and processed in an encrypted format to maintain security and privacy. For the purpose of content notation and/or tampering detection, it is necessary to perform data hiding in these encrypted videos. In this way, data hiding in encrypted domain without decryption preserves the confidentiality of the content. In addition, it is more efficient without decryption followed by data hiding and re-encryption. A novel scheme of data hiding directly in the encrypted version of H.264/AVC video stream is proposed, 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.

Keywords : Data Hiding, Encrypted Domain, H.264/AVC, Codeword Substituting.

I INTRODUCTION

CLOUD computing has become an important technology trend, which can provide highly efficient computation and large-scale storage solution for video data. Given that cloud services may attract more attacks and are vulnerable to untrustworthy system administrators, it is desired that the video content is accessible in encrypted form. The capability of performing data hiding directly in encrypted H.264/AVC video streams would avoid the leakage of video content, which can help address the security and privacy concerns with cloud computing [1]. For example, a cloud server can embed the additional information (e.g., video notation, or authentication data) into an encrypted version of an H.264/AVC video by using data hiding technique. With the hidden information, the server can manage the video or verify its integrity without knowing the original content, and thus the security and privacy can be protected. In addition to cloud computing, this technology can also be applied to other prominent application scenarios. For example, when medical videos or surveillance videos have been encrypted for protecting the privacy of the people, a database manager may embed the personal information into the corresponding encrypted videos to provide the data management capabilities in the encrypted domain. Till now, few successful data hiding schemes in the encrypted domain have been reported in the open literature. With the increasing demands of providing video data security and privacy protection, data hiding in encrypted H.264/AVC videos will undoubtedly become popular in the near future. Obviously, due to the constraint of the underlying encryption, it is very difficult and sometimes impossible to transplant the existing data hiding

algorithms to the encrypted domain. To the best of our knowledge, there has been no report on the implementation of data hiding in encrypted H.264/AVC video streams. The proposed scheme can achieve excellent performance in the following three different prospects.

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

II. RELATED WORK

W. Hong, T. S. Chen, and H. Y. Wu have proposed that most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain[6]. The five MSBs of each pixel of the decrypted image will be identical to those of the cover image. According to the data-hiding key, it is easy for the data hider to reversibly embed data in the encrypted image. Thus the data hider can benefit from the extra space Emptied out in previous stage to make data hiding process effortless.

In the field of video, W. Puech , Z. Erkin, M. Barni, S. Rane proposed SE of H.264 video is proposed by doing frequency domain selective scrambling, DCT block shuffling and rotation. It performs SE by pseudo-randomly inverting sign of DCT coefficients in Region of intrest. A scheme for commutative encryption and watermarking of H.264/AVC. Here SE(selective encryption) of some MB header fields is combined with watermarking of magnitude of DCT coefficients but they are not format compliant. SE scheme based on H.264/AVC has been presented on CAVLC and CABAC for I and P frames .This method fulfills real-time constraints by keeping the same bitrate and by generating a completely compliant bit stream[1].

III. SYSTEM ARCHITECTURE

A novel scheme of data hiding in the encrypted version of H.264/AVC videos is presented, which is made up of three subsequent parts, i.e., H.264/AVC video encryption, data embedding and data extraction.It is necessary to perform data hiding in the encrypted videos for the purpose of the content notation and/or tampering detection. Data hiding in encrypted Domain without decryption preserves the confidentiality of the content. Furthermore, it is more efficient without decryption followed by data hiding and re-encryption. A novel scheme of data hiding directly in the encrypted version of H.264/AVC video stream is proposed. By analyzing the property of H.264/AVC codec, the codewords of intra-prediction modes, the codewords of motion vector differences, and the code words of residual coefficient are encrypted with stream ciphers. Then, a data hider may embed additional data in the encrypted domain by using codeword substitution technique, without the knowledge of original video content.

3.1 Encryption of H.264/AVC Video Stream

An H.264/AVC video encryption scheme with good performance including security, efficiency, and format compliance is been 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. Selective encryption in the H.264/AVC compressed domain has been already put forth on context-adaptive variable length coding (CAVLC) and even

on context-adaptive binary arithmetic coding (CABAC). Further improved and enhanced the previous proposed approach by encrypting more syntax elements.

***1) Intra-Prediction Mode (IPM) Encryption*:** According to H.264/AVC standard, there are four different types of intra coding are supported, which are denoted as Intra_4×4, Intra_16×16, Intra_chroma, and I_PCM. Four intra prediction modes (IPMs) are available in the Intra_16 ×16.

2) Motion Vector Difference (MVD) Encryption: Further 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 carried out on the motion vectors, which yields MVD. In H.264/AVC baseline profile, Exp-Golomb entropy coding is used to encode MVD encryption may change the sign of MVD, but does not affect the length of the codeword and satisfies the format compliance.

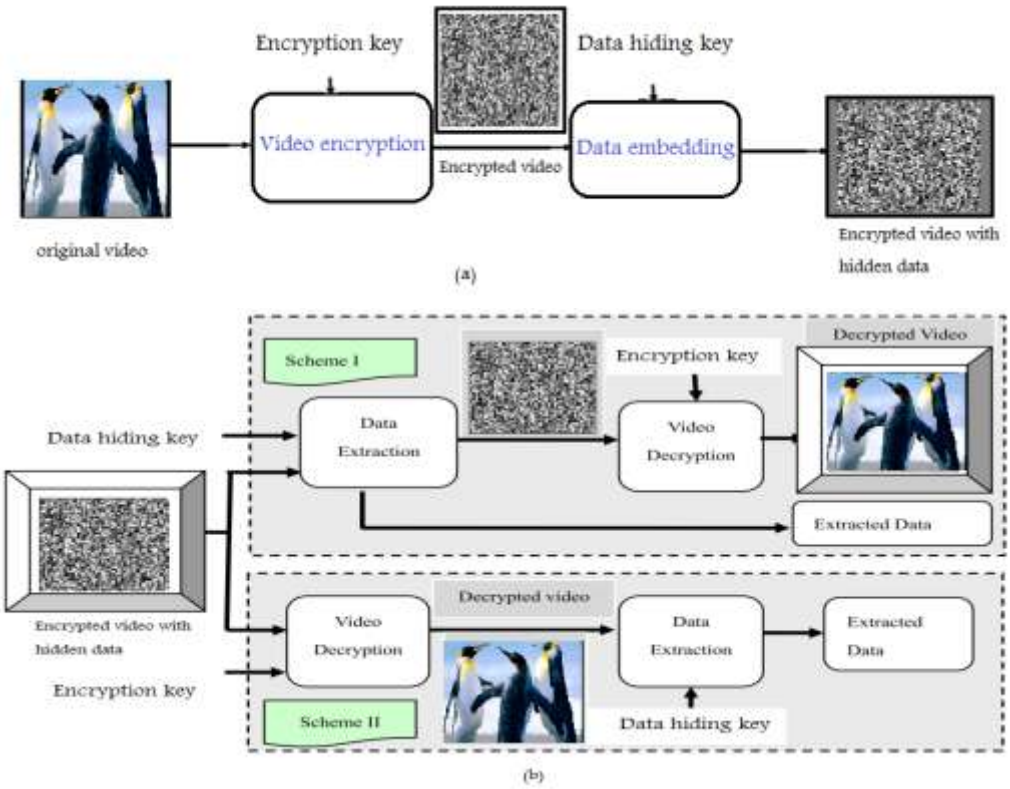

Figure 3.1 Diagram of proposed scheme. (a) Video encryption and data embedding at the sender end. (b) Data extraction and video display at the receiver end in two scenarios

***3) 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 region, a novel method for encrypting the residual data based on the characteristics of codeword.In H.264/AVC baseline profile, CAVLC entropy coding is used to encode the quantized coefficients of a residual block. Each CAVLC codeword can be expressed as the following format:

{Coeff_token,Sign_of_Trailing Ones, Level, Total_zeros, Run_before}

3.2 Data Embedding

In the encrypted bitstream of H.264/AVC, the proposed data embedding is accomplished by substituting eligible codewords. On the other hand, the codewords substitution should fulfill 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. The codewords of Levels which suffix Length is 2 or 3 would be divided into two opposite codespaces denoted as C0 and C1 as shown in Figure. 3.2. The codewords assigned in C0 and C1are associated with binary hidden information "0" and "1". Suppose the additional data that we want to embed is a binary

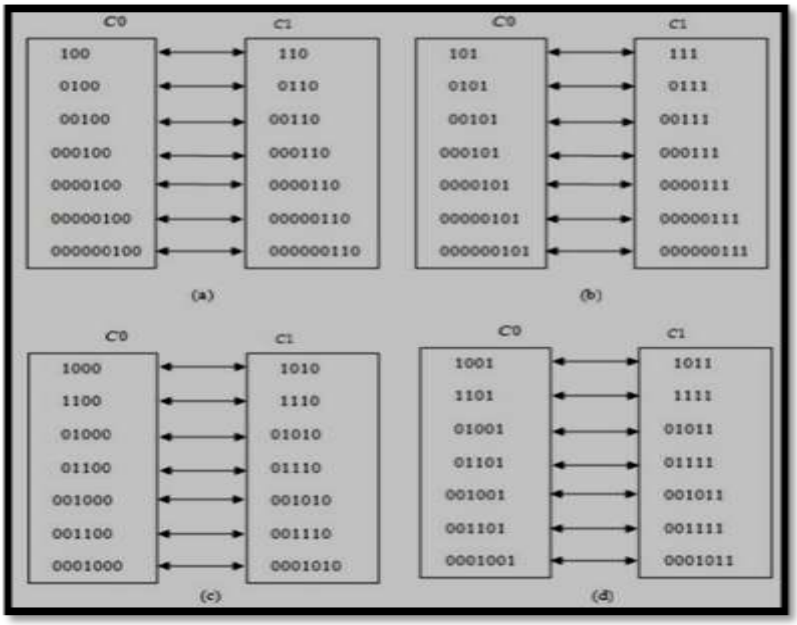

Figure 3.2 CAVLC codeword mapping.

sequence denoted sB={b(i)|i =1,2,⋯,L,b(i)∈ {0,1}}. Data hiding is performed directly in encrypted bit-stream through the following steps.

Step:. In order to enhance the security, the additional data is encrypted with the chaotic pseudo-random sequence P={p(i)|i =1,2,⋯,L,p(i)∈{0,1}}[22] to generate the to-be-embedded sequence W={w(i)|i =1,2,⋯,L,w(i)∈{0,1}}. The sequence Pis generated by using logistic map with an initial value, i.e., the data hiding key. It is very difficult for anyone who does not retain the data hiding key to recover the additional data.

```
Procedure
If(databit==0){
        if(the codword belongs to C0)
         The codeword is unmodified;
         Else if (the codeword belongs to C1)
         The codeword is replaced with the corresponding codeword in C0
            }
If(data  bit==1){
        if(the codword belongs to C1)
         The codeword is unmodified;
         Else if (the codeword belongs to C0)
         The codeword is replaced with the corresponding codeword in C1
           }
```

Figure 3.3. The procedure of codeword mapping

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

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

The detailed procedure of codeword substituting for data hiding is shown in Figure. 3.3. For example, when Level is positive 1 and its suffix Length is 3, then its corresponding codeword is "1000" which belongs to C0 as shown in Figure.3.2. If the data bit "1" needs to be embedded, the codeword "1000" should be replaced with "1010". Otherwise, if the data bit "0" needs to embedded, the codeword "1000" will keep unchanged.

Step4: Choose the next codeword and then go to Step3 for data hiding. If there are no more data bits to be embedded, the embedding process is stopped. Suppose the to-be-embedded data is "1001", the CAVLC codeword of Level parsing from H.264/AVC bitstream is Figure. 3.3. The procedure of codeword mapping

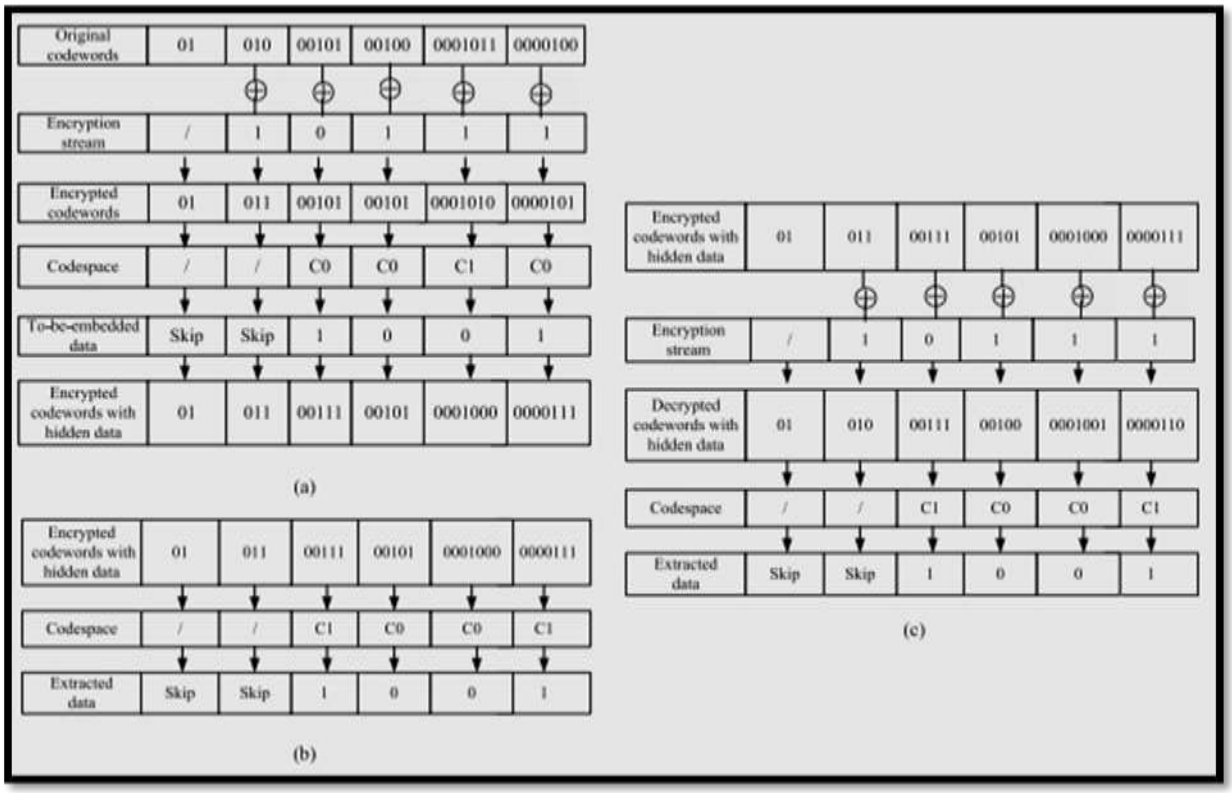

Figure 3.4: An example of data embedding and extraction. (a) Data embedding.(b) Data extraction in encrypted domain. (c) Data extraction in decrypted domain

3.3 Data Extraction

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

***Scheme I*:** Encrypted Domain Extraction.

***Scheme II*:** Decrypted Domain Extraction.

Scheme I: 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 in this

case. In encrypted domain, as shown in Figure. 3.1(b), encrypted video with hidden data is directly sent to the data extraction module. An example of data extraction in encrypted domain is shown in Figure. 3.4(b).

Scheme II: Decrypted Domain Extraction.

In scheme I, both embedding and extraction of the data are performed in encrypted domain. However, in some cases, users want to decrypt the video first and extract the hidden data from the decrypted video. For example, an authorized user, which owned the encryption key, received the encrypted video with hidden data. The received video can be decrypted using the encryption key. That is, the decrypted video still includes the hidden data, which can be used to trace the source of the data. In Figure 3.1(b), the received encrypted video with hidden data is first pass through the decryption module.

The whole process of decryption and data extraction is given as follows.

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

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

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

Step4: the last bit encryption may change the sign of Level. The encrypted codeword and the original codeword are still in the same code spaces.

Step5: Generate the same pseudo-random sequence that was used in embedding process according to the data hiding key. The extracted bit sequence should be decrypted to get the original additional information. An example of data extraction in decrypted domain is shown in Figure. 3.4(c).

IV. CONCLUSION

The existing system just addresses the calculation of smoothness and the process of image recovery and lacks in addressing the data encryption & data embedding process. Time consumption rate is also high when compared to the recent methods developed. The block encryption methods are not robust to noise.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. An algorithm is used to embed additional data in encrypted H.264/AVC bit stream, which consists of video encryption, data embedding and data extraction phases.The data-hider can embed additional data into the encrypted bit stream using codeword substituting, even though he does not know the original video content. Since data hiding is completed entirely in the encrypted domain, here we can preserve the confidentiality of the content completely.

REFERENCES

[1].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.

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