ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Efficient Steganography in Encoded Video Stream Using Motion Vector Difference

[1]K.Kumaresan, [2]G.S. Rizwana Banu , [3]N. Divya, [4]M. Jayabharathi ,[5]J. Karthikayani
[1,2]Assistant Professor, [3-5] UG scholar
[1-5] Department of Computer science and Engineering, K.S.R College of Engineering, Tiruchengode , Namakkal
Email id: [1]kkumaresanphd@gmail.com,
[2]gsrizwana@gmail.com,[3]divyan812cse@gmail.com,[4]
jayabharathislm02@gmail.com,[5]karthikasmartchamp@gmail.com,

Abstract:- **Digital video sometimes are stored and processed in an encrypted format to maintain privacy and security. For the purpose of content notation, 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 AVI video stream, which includes the following three parts, i.e., AVI video encryption, data embedding, and data extraction. By analyzing the property of AVI codec and the code word of motion vector differences are encrypted with stream ciphers. Then, a data hider may embed additional data in the encrypted domain by using code word 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.**

Keywords -Data hiding, encrypted domain, embedded, AVI, code word substituting.

1.INTRODUCTION

Cloudcomputing has become an important technologytrend, which can provide highly efficient computationand large-scale storage solution for video data. Given thatcloud services may attract more attacks and are vulnerableto untrustworthy system administrators, it is desired that thevideo content is accessible in encrypted form. The capabilityof performing data hiding directly in encrypted H.264/AVCvideo 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 embedded the additional information (e.g., video notation, or authentication data) into an encrypted version ofan 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. In [2], a watermarking scheme in the encrypted domain using Paillier cryptosystem is proposed 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. Note that, several investigations on reversible data hiding in encrypted images are reported in [4]–[8] recently. The encryption is performed by using bit-XOR (exclusive-OR) operation. In these methods, however, the host image is in an uncompressed format. In [9], a robust watermarking algorithm is proposed to embedded watermark into compressed and encrypted JPEG2000 images.

II. EXISTING SYSTEM

In existing system, the **Motion Vector Difference (MVD)** Encoding is carried out as follows. In order to protect both texture information and motion information, not only the IPMs but also the motion vectors should be encoded. In avi file, motion vector prediction is further performed on the motion vectors, which yields MVD. The values of MVDs are taken.

For Data Embedding:In the encrypted bitstream of avi frames, the proposed data embedding is accomplished by substituting

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

A novel scheme of data hiding in the encrypted version of AVI videos is presented, which includes three parts, i.e., AVI video encryption, data embedding and data extraction. The content owner encrypts the original AVI 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 decrypted version. In contrast to the existing technologies [10]–[11] discussed above, the proposed scheme can achieve excellent performance in the following three different prospects.

- Both original video and raw data is retrieved.
- The final video is both in encrypted format or original format.
- Perturbing the raw data is carried out and encryption mechanism of raw text data is also carried out for additional security.
- The operation may be carried out in two types. A) First data extraction followed by video decoding or B) Video decoding followed by data extraction.

III. PROPOSED SYSTEM

In proposed system, all the existing system implementation is carried out. In addition, the given raw data is perturbed first, then encrypted with 3DES encryption and addition secure key is also embedded in the message. Then the data is embedded in video file. During decryption, the original video file as well as the decrypted data is retrieved. Then the data is decrypted and the perturbed data is found out. Then the original raw message is retrieved. In this study, an AVI video encryption scheme with good performance including security, efficiency, and format compliance is proposed. By analyzing the property of AVI codec, three sensitive parts (i.e., IPMs, MVDs, and residual coefficients) are encrypted with stream ciphers. Compared with the proposed encryption algorithm is performed not during AVI encoding but in the AVI compressed domain. In this case, the bitstream will be modified directly. Selective encryption in the AVI compressed domain has been already presented on context-adaptive variable length coding (CAVLC) and context-adaptive binary arithmetic coding (CABAC). In this study, 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 AVI compliant and can be decoded by any standard-compliant AVI decoder, but the encrypted video data is treated completely different compared to plaintext video data.

A. Encryption of AVI 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 bit stream 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.

1. Intra-Prediction Mode (IPM) Encryption:

According to AVI standard, the following four types of intra coding are supported, which are denoted as Intra_4 × 4, Intra_16× 16, Intra chroma, and I_PCM. Here, IPMs in the Intra_4 × 4 and Intra_16 × 16 blocks are chosen to encrypt. Four intra prediction modes (IPMs) are available in the Intra_16 × 16. The IPM for Intra_16 × 16 block is specified in the mb_type (macroblock type) field which also specifies other parameters about this block such as coded block pattern (CBP).

2. 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 AVI, motion vector prediction is further performed on the motion vectors, which yields MVD. In AVI baseline profile, Exp-Golomb entropy coding is used to encode MVD. The codeword of Exp-Golomb is constructed as [M zeros] [INFO], where I N FO is an M-bit field carrying information.

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 AVI 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:

{Co- eff token, Sign of TrailingOnes, Level, Total zeros, Run before}

B. Data Embedding

In the encrypted bitstream of AVI, 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.

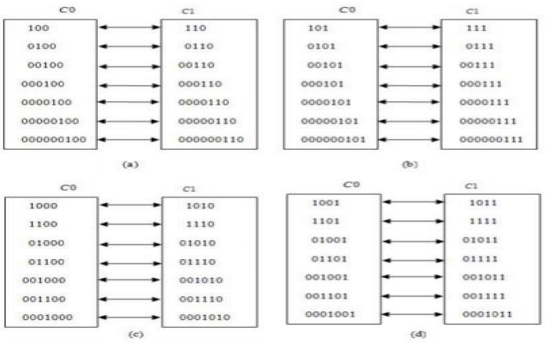

Fig. 2, CAVLC codeword mapping. (a) suffix Length = 2& Level > 0. (b) suffix Length = 2& Level > 0. (c) suffix Length = 3& Level > 0. (d) suffix Length = 3& Level < 0

Procedure of Code word Mapping:

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

C. Data 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.

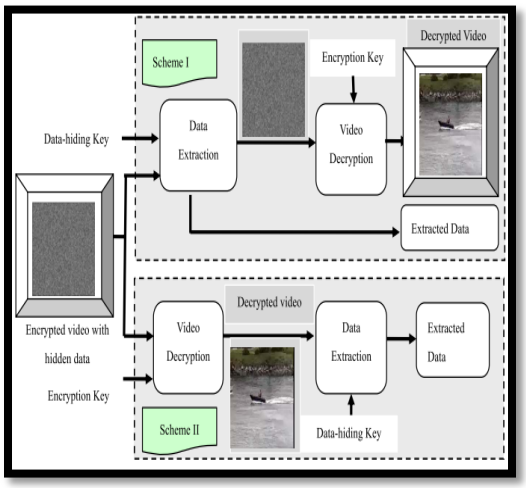

Fig 3. Data extraction and video display at the receiver end in two scenarios.

In encrypted domain, encrypted video with hidden data is directly sent to the data extraction module, and the extraction process is given as follows.

Step1:

The codewords of Levels are firstly identified by parsing the encrypted bitstream.

Step2:

If the codeword belongs to codespace C0, the extracted data bit is "0". If the codeword belongs to codespace 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. Then the extracted bit sequence could be decrypted by using P to get the original additional information. Since the whole process is entirely operated in encrypted domain, it effectively avoids the leakage of original video content.

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 theencryption 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.

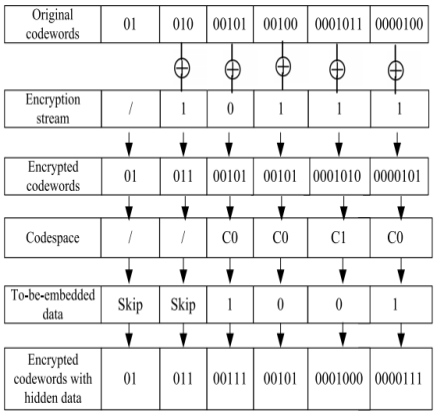

Fig 4: Data embedding

Step1:

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

Step2:

The code words of IPMs, MVDs, Sign of TrailingOnes and Levels are identified by parsing the encrypted bitstream.

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 plaintext. 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.

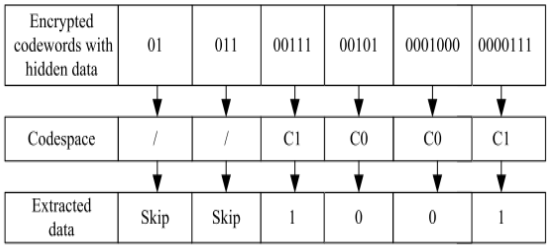

Fig 2.2 Data extraction in encrypted domain

Step4:

The last bit encryption may change the sign of Level. Encrypted codeword and the original codeword are still in the same codespaces. If the decrypted codeword of Level belongs to codespace C0, the extracted data bit is "0". If the decrypted codeword of Level belongs to codespace C1, the extracted data bit is "1".

Step5:

Generate the same pseudo-random sequence P 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.

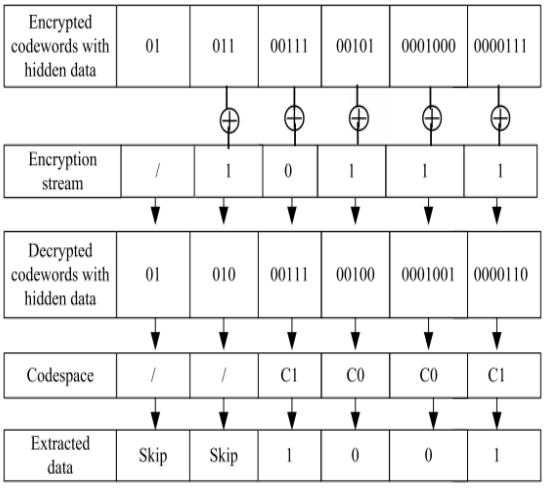

Fig 2.3 Data extraction in decrypted domain

IV. CONCLUSION

Data hiding in encrypted media is a new topic that has started to draw attention because of the privacy preserving requirements from data management. In this paper,an algorithm to embed addition data in encrypted

AVI bit stream is presented,which consider 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 directly performed in the compressed and encrypted, .i.e., it does not require decrypting or partial decompression of the video compression of the video stream thus making it ideal for real-time video applications. The data- hider can embed additional data into the encrypted bit stream using code substituting method. 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 AVI. Experimental results have shown that the proposed encryption and data embedding scheme can preserves file-size, whereas the degradation in video quality caused by data hiding is quite small.

Reference

[1] W. J. Lu, A. Varna, and M. Wu, "Secure video processing: Problems and challenges", in Proc. IEEE Int. Conf. Accost, 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 homomorphicencrypted domain and its application in image watermarking", in Proc. 14th Inf. Hiding Conf., Berkeley, CA, USA, 2012, pp. 1–15.

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

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

[7] [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.

[8] [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] [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.

[10] [10] S. G. Lian, Z. X. Liu, and Z. Ren, "Commutative encryption andwatermarking in video compression",IEEE Trans. Circuits Syst. VideoTechnol., vol. 17, no. 6, pp. 774–778, Jun. 2007.

[11] [11] S. W. Park and S. U. Shin, "Combined scheme of encryption and watermarking in H.264/scalable video coding (SVC)", New DirectionsIntell. Interact. Multimedia, vol. 142, no. 1, pp. 351–361, 2008.

[12] [12] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard", IEEE Trans. Circuits Syst.Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.

[13] [13] S. G. Lian, Z. X. Liu, Z. Ren, and H. L. Wang, "Secure advanced video coding based on selective encryption algorithms", IEEE Trans.Consumer Electron., vol. 52, no. 2, pp. 621–629, May 2006.