ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Data Hiding MPEG-4/AVC Video Streams

[I]T.R Pavan Kumar, [II]S.Narasimhulu , [III]C.H.Lawrence Dheeraj
[I,II,III]Assistant Professor, Dept. of CSE, S.V College of Engineering, Tirupati, AP, India

Abstract
In this paper, a novel scheme of data hiding directly in the encrypted version of MPEG-4/AVC video stream is proposed, which includes the following three parts, i.e., MPEG-4/AVC video encryption, data embedding, and data extraction. By analysing the property of MPEG-4/AVC codec, the code words of intraprediction modes, the code words of motion vector differences, and the code words 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 con- tent. 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 pre- served even after encryption and data embedding. Experimental results have demonstrated the feasibility and efficiency of the proposed scheme

Keywords
Data, Hiding, Embedding, MPEG-4/AVC, Encryption, Decryption

I. Introduction

As said the above mentioned works have been focused On image. With the increasing demands of providing video Data security and privacy protection, data hiding in encrypted MPEG-4/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

MPEG-4/AVC video streams. Only few joint data-hiding and Encryption approaches that focus on video have been proposed. For example, Amplitudes are during MPEG-4/AVC compression, the Intra-prediction mode (IPM), and motion vector difference (MVD) And DCT coefficients' signs are encrypted, while DCT coefficients' watermarked adaptively. a combined Scheme of encryption and watermarking is presented,

This can provide the access right as well as the authentication of video content simultaneously. The IPMs of 4 × 4 luminance Block, the sign bits of texture, and the sign bits of MVDs Are encrypted, while IPM is used for watermarking. However, the watermarked bit stream is not fully format-compliant as a Result a standard decoder may crash since it cannot parse a Watermarked bit stream. Concretely, the value "−2" of IPM does not exist in the actual standard. In summary, in the Existing related technologies encryption and data embedding are implemented almost simultaneously during MPEG-4/AVC compression process. However, to meet the aforementioned application requirements, It's necessary to perform data hiding directly in the encrypted domain. In addition, the approaches in and do not operate on the compressed bit stream. That is, encryption and watermark embedding are

accomplished in the encoding process, while decryption and watermark detection are completed in the decoding process The compression/decompression cycle is time-consuming and hampers real-time implementation. Besides, encryption and watermark embedding would lead to increasing the bit-rate of MPEG-4/AVC bitstream.Therefore, it becomes highly desirable to develop data hiding algorithms that work entirely on encoded bit stream in the encrypted domain However, and 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. Based on the analysis given above, we propose a novel Scheme to embed secret data directly in compressed and then encrypted MPEG-4/AVC bit stream. Firstly, by analysing the property of MPEG-4/AVC codec, the code words of IPMs, the code words of MVDs, and the code words of residual coefficients are encrypted with a stream cipher. The encryption algorithm is combined with the Exp-Golomb entropy coding and Context-adaptive variable-length coding (*CAVLC*) which keeps the codeword length unchanged. Then, data hiding in the encrypted domain is performed based on anovel codeword substituting scheme. In contrast to the existing technologies discussed above, the proposed scheme can achieve excellent performance in the following three different prospects.

- The data hiding is performed directly in encrypted MPEG-4/AVC video bit stream.
- The scheme can ensure both the format compliance and the strict file size preservation.
- The 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.

The remainder of the paper is organized as follows. In Section II, we describe the proposed scheme, which includes three parts, i.e., MPEG-4/AVC video encryption, data embedding and data extraction. Experimental results are presented in Section III. Discussion is shown in Section IV. Finally in Section V, conclusion is drawn.

II. Proposed Work

This section describes the proposed scheme, the phases of data hiding in video and their algorithms. Since a video is created from a sequence of still images called frames, the same technique of image data hiding can be extended to the concept of video data hiding The cover media, here the video, is initially divided into

multiple frames. Each of these frames is an image. The frame is encrypted using any one of the convenient encryption mechanisms with a frame key. The frame key is symmetric. That it, the sender and receiver uses the same key in encryption and decryption. After the data to be embedded is accepted, it is encrypted using a data key. The encrypted data in binary is embedded into the frame using algorithms specified in. At the receiver side, initially the frame is decrypted with the frame key and the data is extracted using algorithms in. This data is then decrypted using the same data key

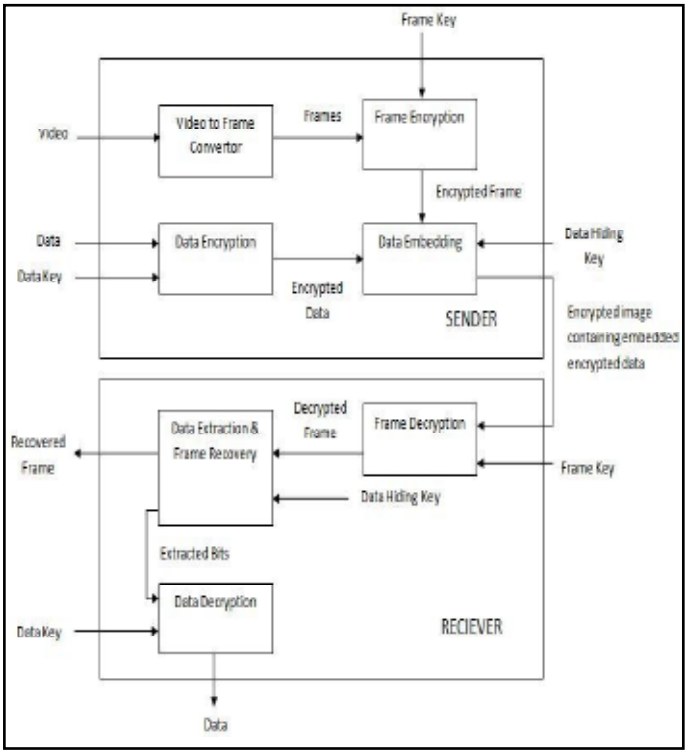

Fig. 2 : Sketch of the proposed scheme

A. Algorithms

1. SNDR_MAIN (Target Image, Cover Image)

This is the main function in our algorithm. This function will also be used in the sender side and will call other modules of our algorithm.
Input: This function will take Target Image and Cover Image as input.
Output: It will output the encoded stego-image.

2. RESIZE (PICTURE, SIZE)

This function is used in the algorithm to resize an image to obtain an image of the desired size from the input image.
Input: This function will take the image, which has to be resized along with the desired image size, which is to be obtained after resizing.
Output: This function outputs an image of desired size.

3. ZOOM (PICTURE, SIZE)

This function is used in the algorithm to zoom an image to obtain an image of the double size of the input image using row-column duplication technique.
Input: This function will take the image, which has to be zoomed along with the desired image size, which is to be obtained after zooming.
Output: This function outputs an image of desired size.

4. SNDR_ENC (PICTURE_1, PICTURE_2)

This function is used in the algorithm to encrypt an image with the help of another image to obtain an encrypted image.
Input: This function will take the cover image (PICTURE_1) [Resized and Zoomed 4x] in which another image will be hidden i.e. the target image (PICTURE_2).
Output: This function will output the stego-image as the final image.

5. RCVR_MAIN (STEGOIMAGE)

This is the main function in our algorithm. This function will be used in the receiver side and will call other modules of our algorithm.
Input: This function will take StegoImage as input.
Output: It will output the decoded Image.

6. RCVR_DCD (PICTURE_1)

This function is used in the algorithm to decrypt an image with the help of a key to obtain the final image.
Input: This function will take the StegoImage (PICTURE_1) in which another image is hidden.
Output: This function will output the Target Image as the final image.

III. Test Result

A. Complexity analysis of the stated algorithm

In case of space complexity at the sender end, for cover image dimension mXn, after the zooming operation, the size becomes 2mX2n. So, for storing this zoomed image, space required=2m*2n=4mn, which is O(mn).The target image is then encrypted and stored in this zoomed image, which does not alter the space complexity. In the receiver end the decryption algorithm performs image scan in row wise order and generate the target image. Thus the time complexity order becomes O(mn). In case of space complexity at the receiver end, if the received
Image size is mXn, then the Final Image is (m/2Xn/2). So, for storing the Final Image space required=(m/2)*(n/2)=mn/4,which is O(mn).

B. Test Results

Sender End:

Fig. 1: Cover Image (300x280)

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

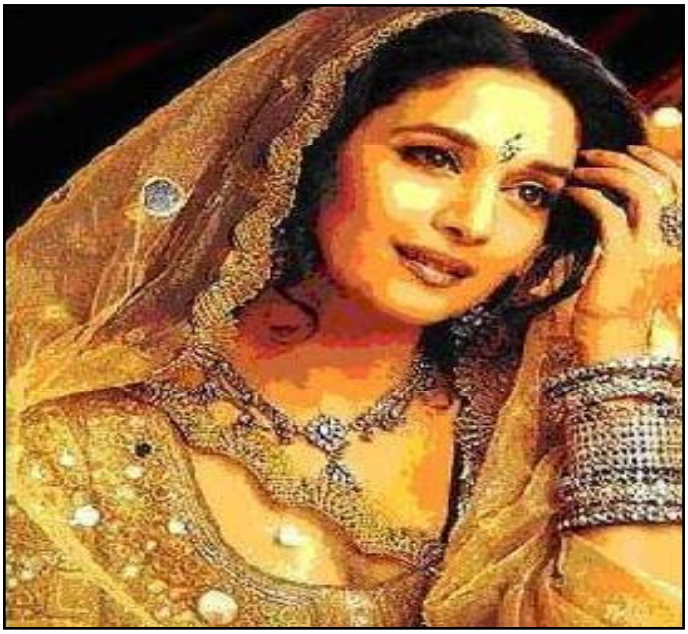

Fig. 2: Targe Image(513x420)

Fig. 3: Stego Image (1026x840) Receiver end:

Fig. 4: Final Image (513x420)

IV. Conclusions

The main advantage of our algorithm is that the final image can be derived only from the StegoImage. The original cover image is not needed for decoding the stego image. This provides less network transmission overhead as well as less scope of suspicion for the network intruder. Moreover this algorithm is free from size constrains i.e. it performs well on any size of the cover image or target image. The fourth pixel value of every quadrant of the stego image is free and using LSB modification can use it for transmission of additional data or DCT based method or any other
method.

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.*

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

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

[10] *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.*

[11] *S. W. Park and S. U. Shin, "Combined scheme of encryption and watermarking in MPEG-4/scalable video coding (SVC)," New Directions Intell. Interact. Multimedia, vol. 142, no. 1, pp. 351–361, 2008.*

[12] *T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the MPEG-4/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.*

[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.

[14] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of MPEG-4/AVC by selective encryption of CAVLC and CABAC for I and P frames," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 5, pp. 565–576, May 2011.

[15] M. N. Asghar and M. Ghanbari, "An efficient security system for CABAC bin-strings of MPEG-4/SVC," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 3, pp. 425–437, Mar. 2013.

[16] T. Stutz and A. Uhl, "A survey of MPEG-4 AVC/SVC encryption," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 3, pp. 325–339, Mar. 2012.

[17] Advanced Video Coding for Generic Audiovisual Services, ITU, Geneva, Switzerland, Mar. 2005

[18] J. G. Jiang, Y. Liu, Z. P. Su, G. Zhang, and S. Xing, "An improved selective encryption for MPEG-4 video based on intra prediction mode scrambling," J. Multimedia, vol. 5, no. 5, pp. 464–472, 2010.

[19] I. E. G. Richardson, MPEG-4 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia. Hoboken, NJ, USA: Wiley, 2003.

[20] D. K. Zou and J. A. Bloom, "MPEG-4 stream replacement watermarking with CABAC encoding," in Proc. IEEE ICME, Singapore, Jul. 2010, pp. 117–121.

Authors Profile

T.R.Pavan Kumar working as assistant professor in Sv engineering College Tirupati andrapradesh. He completed his M.Tech in Computer Science and Engineering at Sri indu Engineering and technology, telangana, India. He has 5 years of teaching experience His interest includes Data Mining, Network Security and cloud computing

S.Narasimhulu working as assistant professor in Sv engineering College Tirupati andrapradesh. He completed his M.Tech in Computer Science and Engineering at Siddharth Institute of Engineering and Technology Puttur, Andhra Pradesh, India. His interest includes Data Mining, Network Security

C.H.Lawrence Dheeraj completed his M.Tech in Computer Science and Engineering at Seshachala Institute of Technology Puttur, Andhra Pradesh, India. His interest includes Data Mining, Network Security and cloud computing