ISSN 23499842(Online), Volume 1, Special Issue 2(ICITET 15), March 2015
International Journal of Innovative Trends and Emerging Technologies

ENHANCED HIDING IN ENCRYPTED VIDEO USING PAILLIER TECHNIQUES

D.ANBUSELVI[1], N.INDHU[2], .ANGELIN BUELAH[3]
[1,2]Student, [2]Faculty, [1,2,3]Department of Information Technology,
[1,2,3]Anand Institute of Higher Technology, Chennai

Abstract -Digital video sometimes need to be stored and processed in an encrypted format to maintain security and privacy. Data hiding techniques can be used to embed a secret message and secret image into a video bit stream for copyright protection, access control and transaction tracking. Some data hiding techniques to assess the quality of video in the absence of the original reference. The quality is estimated based on computing the degradations of the extracted hidden message. Data hiding is also used for concealment in applications of video transmission. Edge orientation information and number of bits of a block are hidden in the bit streams processed in an encrypted format to maintain security and privacy. Futhermore, video file size is strictly preserved even after encryption and data embedding. Experimental results have demonstrated the feasibility and efficiency of the proposed scheme.

Index Terms-Data hiding, paillier techniques, encrypted domain.

I.Introduction

CLOUD computing has become an important technology trend, which can provide highly efficient computation and large-scale storage for video data. Given that cloud services may attract and more attracts 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.267/AVC video streams would avoid the leakage of video content, which can help address the security and privacy concerns with cloud computing.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. This paper proposes two data hiding approaches using compressed MPEG video. The first approach hides message bits by modulating the quantization scale of a constant bit rate video. A payload of one message bit per macro block is achieved. The regression model is then used by the decoder to predict the values of the hidden message bits with very high prediction accuracy. The second approach uses the flexible macro block ordering feature of H.264/AVC to hide message bits. Macro blocks are assigned to arbitrary slice groups according to the content of the message bits to be hidden. A maximum payload of three message bits per macro block is achieved. The proposed solutions are analyzed in terms of message extraction accuracy, message payload, excessive bit rate and quality distortion. Comparisons with previous work reveal that the proposed solutions are superior in terms of message payload while causing less distortion and compression overhead.

II Proposed Scheme

A novel scheme of data hiding directly in the encrypted version of video stream is proposed, which includes the following three parts, i.e., video encryption, data embedding, and data extraction. By analyzing the property of video code. 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 decryption domain.

A)Frame Selection:

In first , select the video file to hide the Secret image and Data. By using the ffmpeg tool, Video was split up into3 formats. First the video was split up into audio and video separately. Then the video part will be converting into n number of frames. In future these frames are used to hide the image and the data.

B) Data hiding using Visual Cryptography:

In second module, we are selecting any two frames from n number of frames. Selecting the Secret image and converting this image into Grey scale image and further converted into Binary image. By using the Visual Cryptography scheme finally the binary image is split up into two shares.

To hide data into the share, the data is encrypted using Pailier byptosys and by using the steganography technique the cipher text is embedded into the two shares. The Invisible Watermarking technique is used to hide two shares into the selected frames and after the image is hided the all frames are converted into video and mix up with audio and finally video was encrypted using the Base 64 Encoder. Visual cryptography is a secret sharing scheme which uses images distributed as shares such that, when the shares are superimposed, a hidden secret image is revealed. In extended visual cryptography, the share images are constructed to contain meaningful cover images, thereby providing opportunities for integrating visual cryptography and biometric security techniques. In this paper, we propose a method for processing halftone images that improves the quality of the share images and the recovered secret image in an extended visual cryptography scheme for which the size of the share images and the recovered image is the same as for the original halftone secret image. The resulting scheme maintains the perfect security of the original extended visual cryptography system.

C)Extracting the Data:

In third module, public key is received by destination and The Encryption was done by using destination public key and video was transmitted .In receiver system, video was Decrypted and split into frames and extracting the shares and data by selecting the frames which was watermarked .After extracting the Image and the Data, the Data should be Decrypted and the Image received was noisy, so it need to reconstruct the image to get Binary image.

1) 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 to manipulate data in encrypted domain. Data extraction in encrypted domain guarantess the feasibility of our scheme

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 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. 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 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.

Step4: According to Table III, the last bit encryption may change the sign of *Level*. The 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".

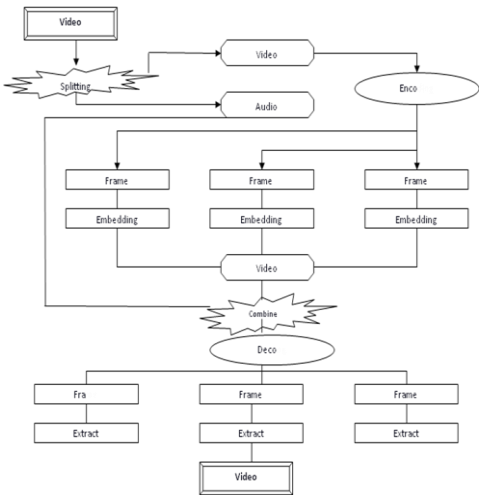
Step5: Generate the same pseudo-random sequence P that was used in embedding process according to the data hiding key. The extracted bit sequence should be decrypted to get the original additional information.

D) Image compression Techniques

When the data is embedded into the image then the required memory is created into the covering media. But if some additional data is required, it is embedded into image then the process of image compression is done. When it is desired to transmit repeated data over bandwidth-constrained channel, it is important to first compress the data and then encode it. Mark

Johnson investigated the innovation of reversing the order of these steps, i.e., first encoding and then compressing. He showed that in certain scenarios his scheme requires no more arbitrariness in the encryption key than the conservative system where compression precedes encryption. Mark Johnson and et.al has examined the possibility of first encrypting a data and then compressing it, such that the compressor does not have knowledge of the key for encryption. The encrypted data can be compacted using dispersed source coding ethics, as the key will be available at the decoder. Wei Liu et.al recommended a lossless compression method for encrypted gray image using progressive decay and rate-compatible punctured turbo codes. In this method they developed resolution progressive compression, which has been shown to have much better coding efficiency and less computational complexity than existing approaches. Wei Liu and et.al observed that lossless compression of encoded sources can be achieved through Slepian-Wolf coding. For encrypted sources such as images, they are trying to improve the compression efficiency. In this paper, he proposed a resolution progressive compression scheme which compresses an encrypted image progressively in oath such that the decoder can monitor a low-resolution report of the image.

III Architectural design



In the above figure, the video can be spitted into audio and converted into frames. Encoding is nothing but an encryption. They are n number of frames choose the any frames in the image. We can embed the frames into the video. After completion of first part, audio is combined with a video. Decode is most important the frame is extracted the data in video.

IV Problem Definition

The existing solutions rely on hiding message bits in discrete cosine transform (DCT) coefficients, motion vectors (MV). Examples of data hiding using DCT coefficients include the use the parity of the quantized coefficients to hide a message. They utilized zero-length codes to insert a dummy value at certain locations to indicate message bits. In MV technique, the phase angles of MV are used to hide message.

CONCLUSION

Enhanced hiding in encrypted video using paillier techique is a new topic that has started to draw attention because of the privacy-perserving 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 descrypting or partial decompression of the video stream thus making it ideal for real-time video applications. The data-hider can embed can embed additional data into the encrypted bitstreams using codeword substituting, even though 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 advantages is that it is fully comliant with the proposed encryption and data embedding scheme can preserve fille-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 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.