International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Data Hiding in H.264/AVC Video Encryption with XOR-ed User Information and Data in File Format

Neenu Shereef

Computer Science and Engineering, ICET, Muvattupuzha, India

Abstract: ***Digital video sometimes needs to be stored and processed in an encrypted format to maintain security and privacy. 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. The enhanced feature added to the system is each user id can be saved inside the video ,so that the uploader of the video can decide who all can access the data within the video.***

Keywords: Data hiding, encrypted domain, H.264/AVC**,** codeword substituting, data decryption

1. 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 . 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. 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. There has been no report on the implementation of data hiding in encrypted H.264/AVC video streams. However, there are some significant challenges for data hiding directly in compressed and encrypted bitstream. The first challenge is to determine where and how the bitstream can be modified so that the encrypted bitstream with hidden data is still a compliant compressed bitstream. 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, here pro-pose a novel scheme to embed secret data directly in compressed and then encrypted H.264/AVC bitstream. Firstly, by analyzing the property of H.264/AVC codec, the codewords of IPMs, the codewords of MVDs, and the codewords 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 a novel codeword substituting scheme. 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.
- 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 purpose of the system is to store datas in a video file in encrypted format. This support all major browser streaming the video upload , the video upload is limited to mp4 only. The system will initially encrypt the video file using the password provided. Each bit of the video file will be XOR-ed with the password provided. The encrypted video will be appended with the data in the encrypted format. The data encrypted by XOR- ing the data and the provided password with it. Decryption of the system will work by providing corresponding password while extracting data and decrypting video. If the password entered is wrong either the data on the video will be change into a format, it will be more complicated to decrypt. The enhanced feature added to the system is each user id can be saved inside the video ,so that

the uploader of the video can decide who all can access the data within the video. The feature will work within the video file and user ids wont saved in database. It is not only data but also text file system can also be uploaded in to the video.

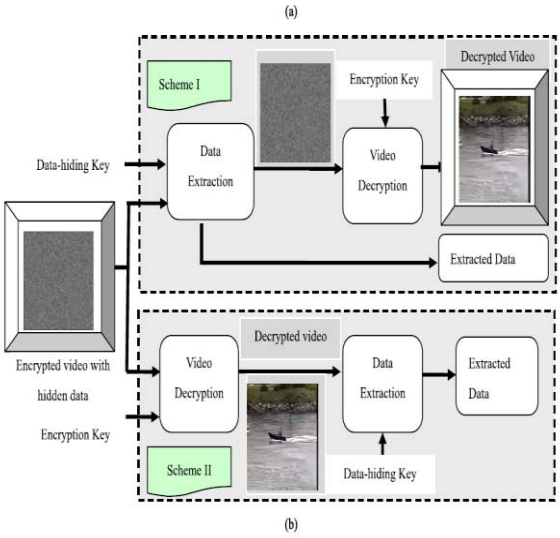

Figure 1: (a)Video encryption and data embedding at the sender end. (b)Data extraction and video display at the receiver end.

1.1. Encryption of H.264/AVC Video Stream

In this paper, here have improved and enhanced the previous proposed approach by encrypting more syntax elements. 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.

1.1.1. Intra-Prediction Mode (IPM) Encryption:

According to H.264/AVC standard, the following four types of intra coding are supported, which are denoted as Intra4_4 , Intra16*16, Intra chroma, and I PCM. Here, IPMs in the Intra44 and Intra16 * 16 blocks are chosen to encrypt. Four intra prediction modes (IPMs) are available in the Intra16*16. The IPM for Intra16*16 block is specified in the mbtype (macroblock type) _eld which also specifies other parameters about this block such as coded block pattern (CBP). In H.264/AVC baseline profile, the mbtype is encoded with the Exp-Golomb code. To maintain standard-compliance of the encrypted bitstream, can encrypt the codeword of an IPM without modifying the CBP. In addition, to keep the codewords length unchanged, the encrypted code-word should have the same size as the original codeword. It can be observed that the combination of CBP is the same in every four lines, and the codewords have the same length in every two consecutive lines.

1.1.2. 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 H.264/AVC, motion vector prediction is further performed on the motion vectors, which yields MVD. In H.264/AVC baseline pro_le, Exp-Golomb entropy coding [19] is used to encode MVD. The codeword of Exp-Golomb is constructed as[M zeros] [1] [I NFO], where I NFO is an M-bit _eld carrying information.The last bit of the codeword is encrypted by applying the bitwise XOR operation with a standard stream cipher, which is determined by an encryption key E Key3.

1.1.3 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 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: Coeffi token, Sign of TrailingOnes, Level, Total zeros, Run before To keep the bitstream compliant, not all syntax elements can be modified during encryption process. For example,Coeff_token, Total zeros, and Run before should remain unchanged . Therefore, residual data encryption can be accomplished by modifying the codewords of Sign of TrailingOnes and Level. The Sign of TrailingOnes is encoded with a single bit. Bit 0 is assigned for +1 and bit 1is assigned for 1. The codeword of Sign of TrailingOnes is encrypted by applying the bitwise XOR operation with a standard stream cipher, which is determined by an encryption key E Key4. The codeword for each Level is made up of a prefix (level prefix) and a suffix (level suffix) as Level codeword = [level prefix], [level suffix]

1.2 Data Embedding

Suppose the additional data that want to embed is a binary sequence denoted as B ={ b(i)|i = 1, 2, , L, b(i)€ {0, 1}}. Data hiding is performed directly in encrypted bitstream through the following steps.

Step1. 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}} to generate the to-be-embedded sequence W = {w(i)|i = 1, 2, ,L, w(i) € {0,1}}. The sequence P is 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.

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

Step3. If current codeword belongs to codespaces C0orC1, the to-be-embedded data bit can be embedded by codeword substituting. Otherwise, the codeword is left unchanged.

1.3 Data Extraction

In this scheme, the hidden data can be extracted either in encrypted or decrypted domain, as shown in Fig 1.Data

extraction process is fast and simple. The extraction in encrypted domain followed by decrypted domain is:

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 have to manipulate data in encrypted domain. Data extraction in encrypted domain guarantees the feasibility of scheme in this case. In encrypted domain, as shown in Fig 6.1, 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 bit-stream.

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.

2) Scheme II:
Step 1: Generate encryption streams with the encryption keys as given in encryption process.

Step 2: The codewords of IPMs, MVDs, Sign of TrailingOnes and Levels are identified by parsing the encrypted bitstream.

Step 3: 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.

Step 4: 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.

Step 5: 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

2. Related Work

2.1 Reversible Data Hiding in Encrypted JPEG Bit-Stream

This proposes a framework of reversible data hiding (RDH) in an en- crypted JPEG bitstream. Unlike existing RDH methods for encrypted spatial-domain images, the proposed method aims at encrypting a JPEG bitstream into a properly organized structure, and embedding a secret message into the encrypted bitstream by slightly modifying the JPEG stream. Identify usable bits suitable for data hiding so that the encrypted bitstream carrying secret data can be correctly decoded. The secret message bits are encoded with error correction codes to achieve a perfect data extraction and image recovery. The encryption and embedding are controlled by encryption and embedding keys respectively. If a receiver has both keys, the secret bits can be extracted by analyzing the blocking artifacts of the neighboring blocks, and the original bitstream perfectly recovered. In case the receiver only has the encryption key, he/she can still decode the bitstream to obtain the image with good quality without extracting the hidden data.The original JPEG bitstream is properly encrypted to hide the image content with the bitstream structure preserved. The secret message bits are encoded with ECC and embedded into the encrypted bitstream by modifying the appended bits corresponding to the AC coefficients. By using the encryption and embedding keys, the receiver can extract the embedded data and perfectly restore the original image. When the embedding key is absent, the original image can be approximately recovered with satisfactory quality without extracting the hidden data.

2.2 Data Hiding in MPEG Video Files Using Multivariate Regression and Flexible Macroblock Ordering

This proposes two data hiding approaches using compressed MPEG video. The first approach hides message bits by modulating the quantization scale of a constant bitrate video. A payload of one message bit per macroblock is achieved. A second order multivariate regression is used to find an association between macroblock-level feature variables and the values of a hidden message bit. 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 exible macroblock ordering feature of H.264/AVC to hide message bits. Macroblocks 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 macroblock is achieved. The proposed solutions are analyzed in terms of message extraction accuracy, message payload, excessive bitrate and quality distortion. In this paper, propose two novel solutions for data hiding. In the first solution, the message bits are hidden by modifying the quantization scale of MPEG video coded with constant bit rates. Features are extracted from individual macroblocks and a second-order regression model is computed. The decoder uses the regression model to predict the content of the hidden message based on macroblock-level feature variables. In the second solution, both constant and

variable bit rate coding are supported. The solution utilizes the Flexible macroblock ordering (FMO) feature of H.264/AVC video for message hiding and extraction.

2.3 An Efficient Text Hiding Approach for H.264/AVC Video

In this paper, an efficient method called Chaos encryption and Twelve Square Substitution Cipher Algorithm used to encrypt the secret hidden text. Chaos encryption algorithm is used to encrypt/decrypt secret text data before/after data embedding/extraction. Shuffle the positions and changing the grey values of image pixels is combined to confuse the relationship between the cipher-image and the plain-image. The data hiding algorithm will convert the text file characters to pixel values using the chaotic shifter which is generated using a logistic map. Larger key space and key sensitiveness are the main advantages. Chaos decryption algorithm is used to decode or reconstruct the secret text data Logistic map is used for generation of chaotic map sequence . It is useful to transmit the secret image through unsecure channel securely which prevents data hacking. The hidden text is encoded and decoded in I frames. The encrypted and decrypted video is done in P frames. Alphabets, digits and special characters are encrypted in twelve-square cipher algorithm . Six 5 by5 matrices are used and each section is arranged in a square1. The letters of the alphabet are placed in each of the 5 by 5 matrices and another six 6 by 7 matrices are arranged in squares for digits and special characters. In this paper, a text is hidden in the encrypted H.264/AVC using chaos encryption method and it is compared with the other encryption method called twelve square substitution cipher algorithm. Double layered security is one of the important factors.

2.4 Enhancing Selective Encryption for H.264/AVC Using Advanced Encryption Standard

Multimedia information availability has increased dramatically with the advent of mobile devices. But with this availability comes problems of maintaining the security of information that is displayed in public. Many approaches have been used or proposed for providing security for information dissemination over networks, protection system classified with more specific as encryption information, and combination between video compression and encryption to increase information security. The strength of the combination between Video compression and encryption science is due to the nonexistence of standard algorithms to be used in video compression and encrypting secret video stream. In this paper proposed a new system of video encryption is presented. The proposed system aim to gain a deep understanding of video data security on multimedia technologies, to investigate how encryption and decryption could be implemented for real time video applications, and to enhance the selective encryption for H.264/AVC.

3. Conclusion

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 domainAnother advantage is that it is fully compliant with the H.264/AVC syntax.

4. Acknowledgment

The Author would like to thank Mrs Liyamol Aliyar Head of Department, Department of Information Technology, Ilahia College of Engineering and Technology, Muvattupuzha for her moral and technical support.

References

[1] M. N. Asghar and M. Ghanbari, “An efficient security system for CABAC bin-strings of H.264/SVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 3, pp. 425–437, Mar. 2013
[2] T. Shanableh, “Data hiding in MPEG video files using multivariate regression and flexible macroblock ordering,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 455–464, Apr. 2012.
[3] W. Hong, T. S. Chen, and H. Y. Wu, “An improved reversible datahiding in encrypted images using side match,” IEEE Signal Process. Lett., vol. 19, no. 4, pp. 199–202, Apr. 2012.
[4] X. P. Zhang, “Reversible data hiding in encrypted image,” IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.
[5] Z. Shahid, M. Chaumont, and W. Puech, “Fast protection of H.264/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.
[6] 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. 351361, 2008.
[7] 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.

Author Profile

Neenu Shereef received the Bachelor of Technology degree in Computer Science from Mahatma Gandhi University, Kerala. She is currently doing Master of Technology degree in Computer Science and Engineering with Specialization in Information Systems from Mahatma Gandhi University, Kerala.