International Journal of Engineering Research and Reviews ISSN 2348-697X (Online)
Vol. 3, Issue 2, pp: (66-70), Month: April - June 2015, Available at: www.researchpublish.com

Digital Copyright Protection of H.264 Video and Encrypted Scheme

[1]Renuka, [2]Thanuja E

[1] student, Electronics and communication, AMC engineering college Bangalore, Karnataka, India
[2] Assistant professor, Electronics and communication, AMC engineering college Bangalore, Karnataka, India

Abstract: **Nowadays everywhere we are facing problem of copyright through online or any other sites so to avoid copyright for specially video here is new method by using codeword mapping algorithm. 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 this algorithm we are defined three different phases to protect from copyright , first one is we need to take AVI video then encrypted using H.264/AVC encrypted scheme then second phase is we need to hide secret key to that encrypted video ,this will be done by authority person he will be not aware of the original video itself so here we can say video is secured ,after once secret key is inserted to encrypted video at receiver he can extract video by two method ,next comes in third phase first user can extract video then get back hidden key or first get hidden key then extract original video .This module will give very low encoding bit rate compare to other format and video quality will remain same at the receiver .we can apply this for any real time application where we need video security purpose like patient personal information can be secured in video and military purpose to send secret information through video.**

Keywords: **Data embedding encrypted scheme, H.264/AVC, code word mapping algorithm.**

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. 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.

A. Problem Statement

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.

II. EXISTING SYSTEM

In the existing related technologies, encryption and data embedding are implemented almost simultaneously during H.264/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 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 H.264/AVC bit stream.

Therefore, it becomes highly desirable to develop data hiding algorithms that work entirely on encoded bit stream in the encrypted domain. However, there are some significant challenges for data hiding directly in compressed and encrypted bit stream.

Limitations:

- 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 application

III. PROPOSED SYSTEM

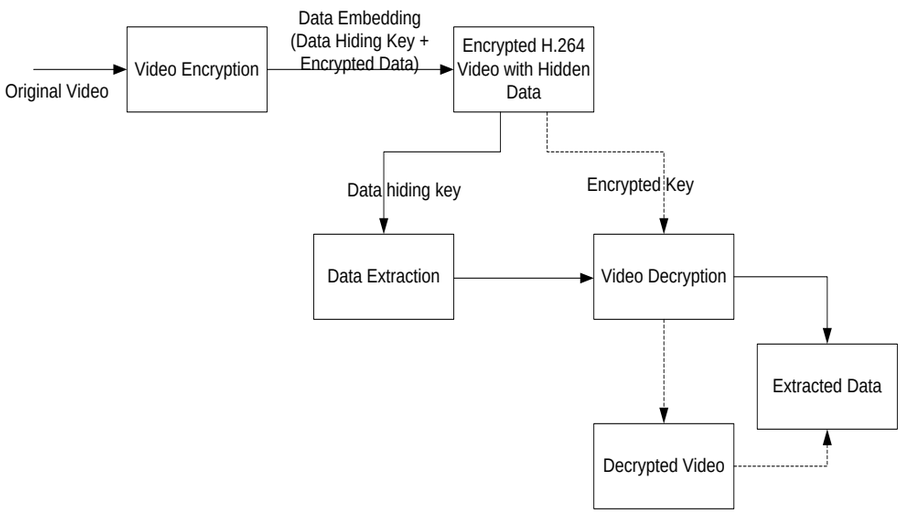
In this section, a novel scheme of data hiding in the encrypted version of H.264/AVC videos is presented, which includes three parts, i.e., H.264/AVC video encryption, data embedding and data extraction. The content owner encrypts the original H.264/AVC video stream using standard stream ciphers with 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 in decrypted version. The diagram of the proposed framework is shown in Fig. 1, where the encryption and data embedding are depicted in part (a), and the data extraction and video decryption are shown in part (b).

A. H.264 AVC Encryption scheme

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. The key issue is then how to select the sensitive data to encrypt. According to the analysis given in [13], it is reasonable to encrypt both spatial information (IPM and residual data) and motion information (MVD) during H.264/AVC encoding.

In this paper, an H.264/AVC video encryption scheme with good performance including security, efficiency, and format compliance is proposed. By analysing the property of H.264/AVC codec, three sensitive parts (i.e., IPMs, MVDs, and residual coefficients) are encrypted with stream ciphers. Compared with [13], the proposed encryption algorithm is performed not during H.264/AVC encoding but in the H.264/AVC compressed domain. In this case, the bit-stream will be modified directly. Selective encryption in the H.264/AVC compressed domain has been already presented on context-adaptive variable length coding (*CAVLC*) [14] and context-adaptive binary arithmetic coding (*CABAC*) [15]. In this paper, we have improved and enhanced the previous proposed approach by encrypting more syntax elements. We encrypt the code words of IPMs, the code words of MVDs, and the code words of residual coefficients. The encrypted bit stream 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 plain-text video data. In fact, performing the format-compliant

B. Block Diagram



C. Data Embedding or Hiding

Although few methods have been proposed to embed data into H.264/AVC bit stream directly [20]–[21], however, these methods cannot be implemented in the encrypted domain. In the encrypted bit stream of H.264/AVC, the proposed data embedding is accomplished by substituting eligible codeword's of *Levels* in Table III. Since the sign of *Levels* are encrypted, data hiding should not affect the sign of *Levels*. Besides, the codeword's substitution should satisfy the following three limitations.

First, the bit stream 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 code words of *Levels* within P-frames are used for data hiding, while the codeword's 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.According to the analysis given above, we can see that there are no corresponding substituted codeword's when *suffix Length* is equal to 0 or 1, as shown in Table III. When *suffix Length* is equal to 0, we cannot find a pair of codeword's with the same size. When *suffix Length* is equal to 1, one codeword also cannot be substituted by another codeword with the same size, since this substitution would change the sign of *Level*. Then the code words of *Levels* which *suffix Length* is 2 or 3 would be

D. Data Extraction

In this scheme, the hidden data can be extracted either in encrypted or decrypted domain, as shown in Fig. 1(b). Data extraction process is fast and simple. We will first discuss the extraction in encrypted domain followed by decrypted domain.

[9] ***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 our scheme in this case.

In encrypted domain, as shown in Fig. 1(b), encrypted video with hidden data is directly sent to the data extraction module, and the extraction process is given as follows.

Step1: The code words of *Levels* are firstly identified by parsing the encrypted bit stream.

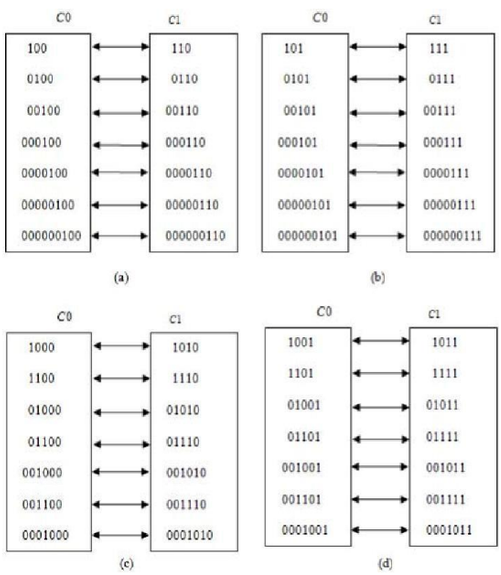
Step2: If the codeword belongs to code space *C*0, the extracted data bit is "0". If the codeword belongs to code space *C*1, the extracted data bit is "1".

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.

Mapping method has been shown in the below figure




```
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.
  }
```

IV. SYSTEM ARCGITECTURE

A. Hardware Requirements

Processors : Pentium IV.
RAM : 64 MB.
Storage : 20GB.
Monitor : 15"
Keyboard : Standard 102 keys

B. Software (tools &technologies) requirements

Platform: Windows 7 or XP
Language : MATLAB
IDE/tool : MATLAB 2013Ra

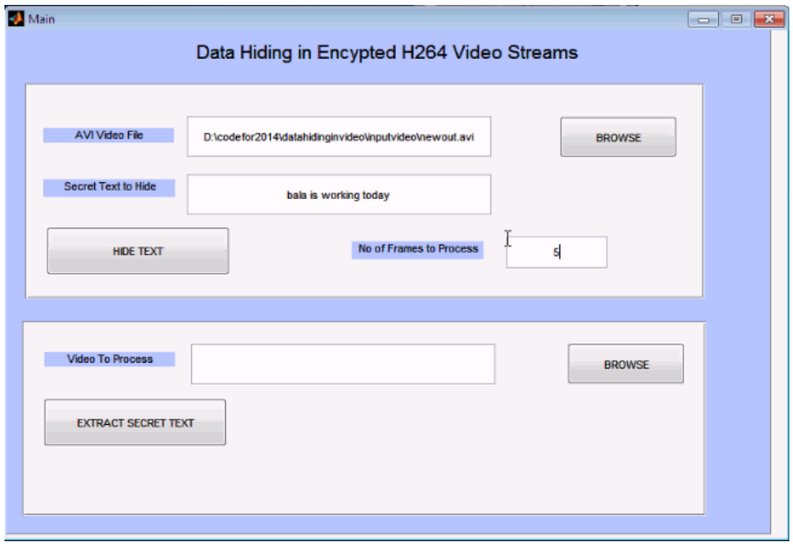
V. EXPERIMENTAL RESULTS

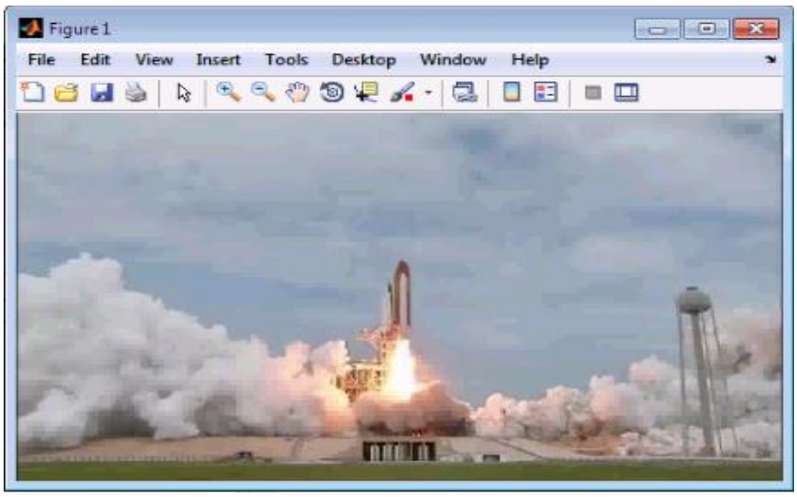
The proposed data hiding scheme has been implemented in the H.264/AVC reference software version JM-12.2. Six well-known standard video sequences (i.e., *Stefan*, *Table*, *Tempete*, *Mobile*, *Hall*, *and News*) in QCIF format (176 × 144) at the frame rate 30 frames/s are used for simulation. The first 100 frames in each video sequence are used in the experiments. The GOP (Group of Pictures) structure is "IPPPP: one I frame followed four P frames".

With experiment simulation results also obtained showed as image output.

INTERPRETATION OF RESULT

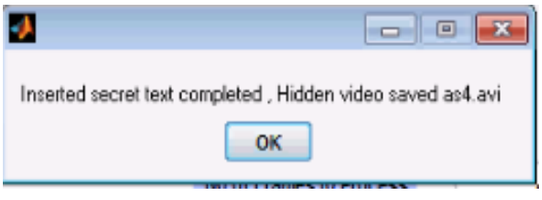
Loading input video



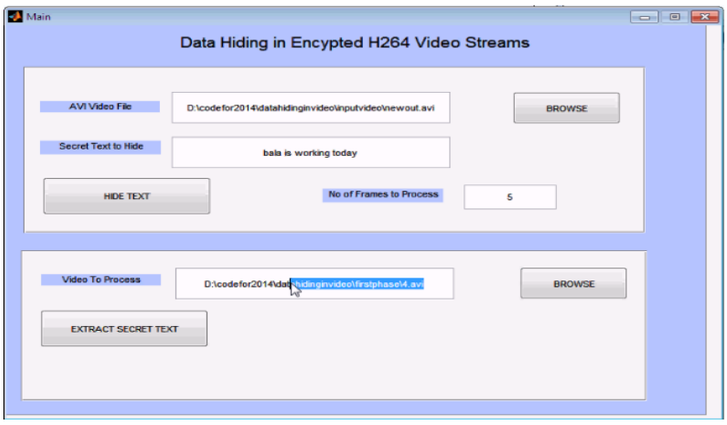


Encrypted input video

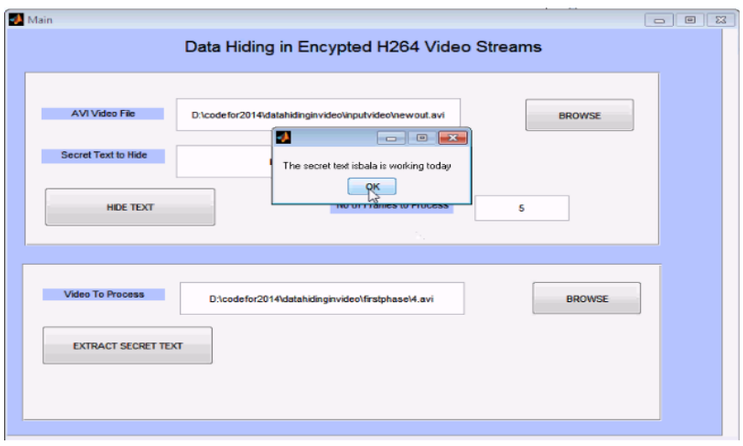
Inserted hidden key Is saved



Loading encryoted video to extract hidden key



Extracting hidden key data from embeded video



VI. CONCLUSION

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. In this paper, an algorithm to embed additional data in encrypted H.264/AVC bit stream 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, 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 advantage is that it is fully compliant with the H.264/AVC syntax. Experimental results have shown that the proposed encryption and data embedding scheme can preserve file-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,” Inproc. 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

[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] Dd 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.