IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 01-05
www.iosrjournals.org

Reversible Data Hiding In Encrypted Video Using Context Free Grammar

Samiksha Velip[1], Dr. J. A. Laxminarayana [2]
[1](*Computer Engineering Department, Goa Engineering College, Farmagud, India*)
[2](*Computer Engineering Department, Goa Engineering College, Farmagud, India*)

Abstract: *In recent years the topic of data hiding has become very popular and a lot of research is being done in this field. The use of computer networks for data transmission has created the need for security. The security plays an important role in transmission of confidential data over internet. So, as a part of improving security in data transmission, we aim to hide the data inside an encrypted video. Thereby, confidentiality of the video as well as the embedded data is maintained. The embedded data can be extracted without any error, and also the cover image restoration is also free from error.*
In this paper, we embed the data in form of context free grammar in an image by using LSB matching technique, so that fast, optimal and lossless steganography is achieved. The proposed method provides total data security and total reversibility, that is, data extraction and image/video recovery.
Keywords: *AES, Context Free Grammar (CFG), LSB, Reversible data hiding (RDH), RSA*

I. Introduction

Data hiding is a group of techniques used to put a secure data in a host media (like images, video) with small deterioration in host and the means to extract the secure data afterwards. For example, steganography can be named as data hiding technique. Steganography is one such pro-security innovation in which secret data is embedded in a cover [1].

Reversible data-hidings insert information bits by modifying the host signal, but enable the exact (lossless) restoration of the original host signal after extracting the embedded information. Sometimes, expressions like distortion-free, invertible, lossless or erasable watermarking are used as synonyms for reversible watermarking. In most applications, the small distortion due to the data embedding is usually tolerable. However, the possibility of recovering the exact original image/video is a desirable property in many fields, like legal, medical and military imaging. Let us consider that sensitive documents (like bank checks) are scanned, protected with an authentication scheme based on a reversible data hiding, and sent through the Internet. In most cases, the watermarked documents will be sufficient to distinguish unambiguously the contents of the documents. However, if any uncertainty arises, the possibility of recovering the original unmarked document is very interesting [7].

Lossless data embedding techniques may be classified into one of the following two categories: Type I algorithms employ additive spread spectrum techniques, where a spread spectrum signal corresponding to the information payload is superimposed on the host in the embedding phase. At the decoder, detection of the embedded information is followed by a restoration step where watermark signal is removed, i.e. subtracted, to restore the original host signal. Potential problems associated with the limited range of values in the digital representation of the host signal, e.g. overflows and underflows during addition and subtraction, are prevented by adopting modulo arithmetic. Payload extraction in Type-I algorithms is robust. On the other hand, modulo arithmetic may cause disturbing salt-and-pepper artifacts. In Type II algorithms, information bits are embedded by modifying, e.g. overwriting, selected features (portions) of the host signal -for instance least significant bits or high frequency wavelet coefficients-. Since the embedding function is inherently irreversible, recovery of the original host is achieved by compressing the original features and transmitting the compressed bit-stream as a part of the embedded payload. At the decoder, the embedded payload- including the compressed bit-stream- is extracted, and original host signal is restored by replacing the modified features with the decompressed original features. In general, Type II algorithms do not cause salt-and-pepper artifacts and can facilitate higher embedding capacities, albeit at the loss of the robustness of the first group

II. Related Work

The previous method can be summarized as the framework in which we are vacating room after encryption (VRAE).

In [1], authors are using technique of Reserving Room Before Encryption (RRBE) using which authors can overcome the drawbacks of existing system i.e. the cover file was decrypted with distortions in it and decrypt the cover without any distortion. In this the sender encrypts the video and data separately, hides the data in encrypted video using LSB technique, while system auto generates the two respective keys. Receiver will need both the keys to extract the data. First, the encryption key is used for decrypting the video then using data hiding key the original data can be extracted.

In this framework, the owner first partitions the video into number of image frames. Some room is reserved in each image in order to hide additional data. Then we encrypt the image using a standard cipher with an encryption key. Here RSA algorithm is used for image encryption. After producing the encrypted image, the owner hands over it to a data hider (e.g. a database manager) and the data hider can embed some auxiliary data into the encrypted image by losslessly vacating some room according to a data hiding key. All the frames are combined to form original video and send it to receiver. Then a receiver can extract the embedded data with the data hiding key and further recover the original image from the encrypted version according to the encryption key.

In [2], 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 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. Furthermore, video file size is strictly preserved even after encryption and data embedding.

In [3], authors have proposed a system that performs reversible data hiding by using the histogram shift operation for RDH. In this system used the spare space for embedding the data by shifting the bins of gray scale values. The embedding capacity measured by the use of number of pixels in peak point. This system has some benefits such as it is simple and has constant PSNR ratio, capacity is high and distortion is very low. This system has some disadvantages such as more time consuming while searching the image number of times.

In [4], the authors are interested in context-free grammars G in which L (G) consists of exactly one string. Given a data string z over a finite alphabet and a context-free grammar G such that *L (G)* = {z}, one can reduce G to a simpler grammar G' for which *L (G')* = {z} and for which certain constraints are satisfied.
One then losslessly compresses z by losslessly compressing the grammar G'. The redundancy performance of this compression algorithm based upon reduced grammars is discussed.
To simplify the grammar the following two constraints on the grammar G are imposed:

1. In the right-hand sides of the production rules of the grammar, there can be no substring of length two that appears in two non-overlapping positions.
2. Every non-terminal symbol of the grammar (except S) must appear at least twice on the right-hand sides of the production rules of the grammar.

In [5], the authors present an improved algorithm which incorporates a length control mechanism into the generation process to produce more and simpler sentences. Experiments demonstrate that besides the benefits for user validation and error location, this algorithm is also helpful in highlighting more errors in some cases.

Purdom provided an algorithm for generating a minimal set of sentences that uses all the rules of a grammar. Authors found out through analysis and experiments that in most cases, Purdom's algorithm produces too few sentences and some of them are long and complex, i.e., with complicated derivation structures. These sentences are not very ideal to construct a test set for the grammar. For this reason, authors propose an improved algorithm which generates more and simpler sentences that might be beneficial to the testing task. The algorithm builds upon Purdom's but differs in integrating a length control mechanism in sentence generation process.

III. Context Free Grammar

Definition 1 (Context Free Grammar): A context free grammar is defined as G = (V, T, P, S), each of which means the following, respectively.

• V is a set of variables.

• T is a finite set of terminate symbols ($T \cap V = \emptyset$).

• P is a finite set of production rules A → α, A ∈ V, α ∈ (V ∪ T)*. Assume that for each A ∈ V, there exists one or more production rule which has A as a left-hand side (They are called rule A's).
• S is a start symbol.
For a context free grammar G, each of V, T, P, S is written as V (G), T (G), P (G), S (G), respectively. For simplicity, the set P (G) of production rules is called grammar G. Any sequence in (V ∪ T)* is called string.

IV. Proposed Methodology

The main aim is to develop a technique in reversible data hiding which will hide secret data in encrypted video which on decryption only intended owner can decipher the hidden message.

The content owner first divide the video in image frames and then compresses the least significant bits (LSB) of the image to create a sparse space to accommodate the additional data. The content owner then encrypts the image frame using an encryption key to produce an encrypted image. Data hider embeds additional data/secret data in the encrypted image using the data hiding key. Additional/secret data is in form of context free grammar which on decryption and parsing only single string can be generated. The content owner then merge all the image frames to form the original video.

At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted video using data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in the video similar to the original version. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original video can be perfectly recovered.

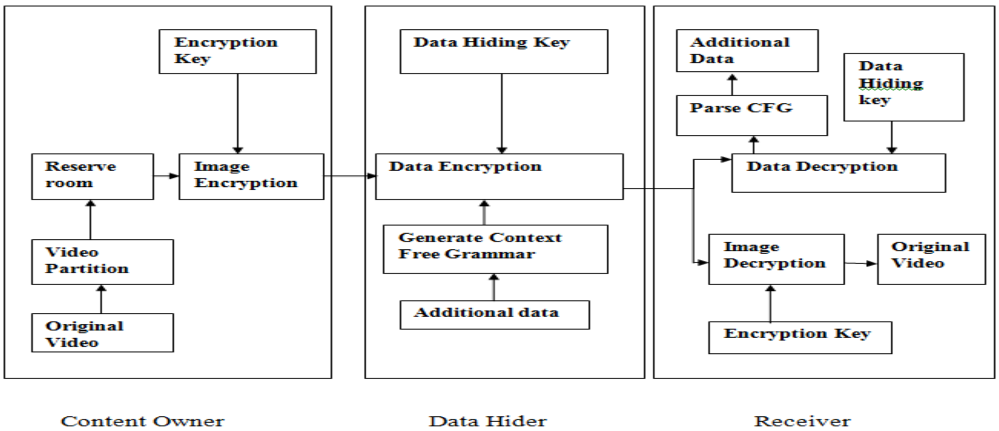

Fig.1 Block diagram of proposed system

A. Encrypted Image Generation

a. Video Partition

At the beginning, original video is divided into number of image frame. Key frames are selected in which we will be used to hide secret data.

b. Image Partition

Image partition step divides original image into two parts **A** and **B**; then, the LSBs of **A** are reversibly embedded into **B** with a standard RDH algorithm so that LSBs of **A** can be used for accommodating messages; at last, encrypt the rearranged image to generate its final version. For example assume the original image C is a gray-scale image with its size M x N, it is divided in to two equal sized images. In this the B part has the smoother area to apply the RDH technique. The LSBs of the pixels of A where the data is hiding is stored.

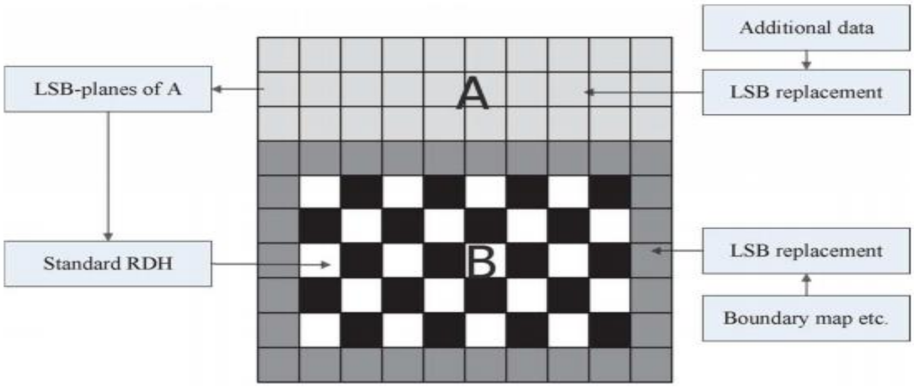

Fig 2: Image Partition using LSB

c. Image Encryption

After image partitioning technique to vacate some space, images frames are encrypted using encryption key. RSA algorithm is used to encrypt image frames. After content owner encrypt the images, this images are given to data hider/ database manager or content owner himself can embed additional data in encrypted images.

B. Data Hiding In Encrypted Image

After image encryption, the sender hands over it to a data hider (e.g., a database manager) and the data hider can embed some auxiliary data.

a. Generation Of Context Free Grammar

Secret data to be embedded in plain text is not secure. In order to prevent hacker from intercepting hidden message, the input message is in form of context free grammar, which when parsed by intended user can intercept the message.

For Example: consider secrete message “reversible data hiding in image “. CFG generated for following message is

S -> N V
N -> P Q
P -> T O
V -> R A
T -> reversible
Q -> hiding
A -> image
R -> in
O -> data

b. Data Hiding

Using the data hiding key above grammar is embedded into encrypted image frames. AES algorithm is used to generate data hiding key. After hiding data, all image frames are combined and send to intended user. Now the final video contains hidden message used as secret communication.

C. Data Extraction And Video Recovery

Since data extraction is completely independent from video decryption, the order of them implies two different practical applications.

a. Extracting Data from Encrypted Video

When the receiver gets the data hiding key, he can decrypt and extract the additional data by directly reading the decrypted version. As the whole process is entirely operated on encrypted domain, it avoids the leakage of original content.

b. Extracting Data From Decrypted Video

If the receiver has the encryption key, then he/she can decrypt images using encryption key and can intercept original images and original video can be recovered.

V. Conclusion

A Reversible data hiding method based on Context Free Grammar (CFG) is proposed to embed the secret data in image/video with high security, imperceptibility and robustness. In the proposed technique the hidden message is in context free grammar, which on parsing generates single and original string back. The security of the data is much more increased due to the use of context free grammar.

References

[1]. Gayatri Pisal, Aparna More, Sonali Pawar, and Ashwini Vishwakarma "Separable Reversible Data Hiding in Encrypted Video File IJRCS - International Journal of Research in Computer Science, Volume: 02 Issue: 04 2015.

[2]. Mr. Ratnakar Kumbhar, Mr. Sagar Yamgar, Ms. Rupali Ghule, Prof. Murkute P.K "Data Hiding in Encrypted H.264/AVC Video Streams By Codeword Substitution" IEEE International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 3, Issue 2, pp: (323-331), Month: April - June 2015

[3]. Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding", IEEE Trans.Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354362, Mar.2006.

[4]. J. Kieffer and E-h. Yang, "Design of context-free grammars for lossless data compression,"Technical Report, Dept. of Electrical & Computer Engineering, University of Minnesota Twin Cities.

[5]. Lixiao Zheng, Duanyi Wu, "A Sentence Generation Algorithm for Testing Grammars", 2009 33rd Annual IEEE International Computer Software and Applications Conference

[6]. Masoud Nosrati, Ronak Karimi, and Mehdi Hariri," Reversible Data Hiding: Principles, Techniques, and Recent Studies, World Applied Programming, Vol (2), Issue (5), May 2012. 349-353.

[7]. Xinpeng Zhang, "Separable Reversible Data Hiding in Encrypted Image", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 7, NO. 2, APRIL 2012

[8]. X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," IEEE Trans. Inform. Forensics Security, vol. 6, no. 1, pp. 53–58, Feb. 2011.

[9]. W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images," IEEE Trans. Image Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.

[10]. M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct.2004.