Secure and Efficient Encrypted Image Based on Reversible Image Transformation

Mr. Vitthal Khandke[1]
[1] Department of Computer Engineering
[1] TSSM's PVPIT, Pune

Abstract- *It is vital to protect the privacy of data with the popularity of outsourcing data to the cloud, and enable the cloud server to easily manage data at the same time. Under such demands, reversible data hiding in encrypted images (RDH-EI) attracts more and more researcher's attention. The system a method of image cryptography based on reversible image transformation. Three parties are involved in the framework including the content owner, the data hider and the recipient. The content owner encrypts the original image using a stream cipher algorithm and uploads cipher text to the server. The data hider on the server divides the encrypted image into three channels and respectively embeds different amount of additional bits into each one to generate a marked encrypted image or the recipient side. Additional message can be extracted from the marked encrypted image and the original image can be recovered without any errors. Since it maintains the excellent property that the original cover can be losslessly recovered after embedded data is extracted while protecting the image content confidentiality. While most of the traditional methods use one criterion to recover the whole image, we propose to do the recovery by a progressive mechanism. Rate distortion of the proposed method outperforms RDH EI methods.*

Keywords- Reversible data hiding (RDH), Image cryptography, Image encryption Criterion.

I. INTRODUCTION

Reversible data hiding (RDH) in images is a technique, by which the original cover can be losslessly recovered after the embedded message is extracted. It is important technique is widely used in medical imagery, military imagery and law forensics, where no distortion of the original cover is allowed. Since RDH has attracted considerable research interest. Nowadays outsourced storage by cloud becomes a more and more popular service, especially for multimedia files, such as images or videos, which need large storage space. To manage the outsourced images, the cloud server may embed some additional data into the images, such as image category and notation information, and use such data to identify the ownership or verify the integrity of images. Obviously, the cloud service provider has no right to introduce permanent distortion during data embedding into the outsourced images [1].

Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain. But, in some applications, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. And it is also hopeful that the original content should be recovered without any error after image decryption and message extraction at receiver side [6].

On the other hand, cloud service for outsourced storage makes it challenging to protect the privacy of image contents. For instance, recently many private photos of Hollywood actress leaked from iCloud. Although RDH is helpful for managing the outsourced images, it cannot protect the image content. Encryption is the most popular technique for protecting privacy [1]. Reversible data embedding, which is also called lossless data embedding, embeds invisible data (which is called a payload) into a digital image in a reversible fashion. As a basic requirement, the quality degradation on the image after data embedding should be low. An intriguing feature of reversible data embedding is the reversibility, that is, one can remove the embedded data to restore the original image. From the information hiding point of view, reversible data embedding hides some information in a digital image in such a way that an authorized party could decode the hidden information and also restore the image to its original [7]. The creates a sparse space to accommodate some additional data by compressing the LSBs of the encrypted image. It is hard to squeeze room by only considering three LSBs of the encrypted images. Instead, chose a half of fourth LSB as the space to carry the data. To further improve the compression ratio, the smooth blocks in the encrypted image, and embed the additional data into the blocks in a sorted order with respect to block smoothness by using local HS. Although the methods in divide the image into patches or groups, the preserved spaces are all acquired by using the LSB modification or compression. As the entropy of encrypted images is maximized, it is difficult to losslessly vacate room after encryption (VRAE) using the above methods. To overcome this drawback, the methods of reserving room before encryption (RRBE) are proposed a large portion of pixels are utilized to estimate the rest before encryption, the additional data is embedded in the encrypted image by operating the estimating errors. the reserving room is obtained by embedding LSBs of some pixels into other pixels. The spare space emptied out is three LSBs of the selected pixels [3]. we are using two frameworks: Framework I "vacating room after encryption (VRAE)" and Framework II "reserving room before encryption (RRBE)."

In the framework 'VRAE," the cloud server embeds data by losslessly vacating room from the encrypted images by using the idea of compressing encrypted images. In the framework "RRBE," the image owner first empties out room by using RDH method in the plain images. After that, the image is encrypted and outsourced to the cloud and the cloud server can freely embed data into the reserved room of the encrypted image. For both frameworks, VRAE and RRBE, the image owner will send a cipher text-formed image to the cloud. However, the cipher texts with the special form of messy codes are easy to cause the attention of the cloud server who may try to dig out information on the encryption users. Therefore, the fact, that the user is outsourcing encrypted images, itself is also a kind of privacy of the user, which should be protected [1].

II. RELATED WORK

The method which consists of three phases: Image encryption, data embedding data extraction/image recovery. In phase I, the sender encrypts the original image into an encrypted image using a stream cipher and an encryption key. In phase II, the data-hider selects and compresses some MSB of the secret image using LDPC codes to generate a spare space, and embeds additional bits into the encrypted image using an embedding key. In phase III, the receiver extracts the secret bits using the embedding key. If he has the encryption key, the original image can be approximately reconstructed via image decryption and estimation. When both the encryption and embedding keys are available, the receiver can extract the compressed bits, and implement the distributed source decoding using the estimated image as side information to perfectly recover the original image [2]. We give following three aspects: encrypted image generation. data hiding in the encrypted image and data extraction and image recovery. For simplicity, we use the grayscale images with 8 bits per pixel. The extension from gray images to color images is straight forward [3]. The framework of the main idea of this method is first to estimate a part of the pixels in an original image using the rest pixels and obtain the estimation errors. Then we encrypt the estimation errors and the rest pixels separately using the encryption key. The data hider then embeds the secret data into the encrypted estimation errors using the data hiding key and scrambles the image using the sharing key.

At the receiver side, the secret data and original image can be extracted and recovered separately by using different security keys [4]. Since losslessly vacating room from the encrypted images is relatively difficult and sometimes inefficient, why are we still so obsessed to find novel RDH techniques working directly for encrypted images? If we reverse the order of encryption and vacating room, i.e., reserving room prior to image encryption at content owner side, the RDH tasks in encrypted images would be more natural and much easier which leads us to the novel framework, "reserving room before encryption (RRBE)". which primarily consists of four stages: generation of encrypted image, data hiding in encrypted image, data extraction and image recovery [5]. The method made up of image encryption, data embedding, and data extraction/image recovery phases [6]. A common approach of high capacity reversible data embedding is to select an embedding area (for example, the least significant bits of some pixels) in an image, and embed both the payload and the original values in this area (needed for exact recovery of the original image) into such area [7].

II. PROPOSED WORK

The proposed scheme is made up of image encryption, block pairing, block transformation and Data embedding, data Removing/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider compresses the least significant bits (LSB) of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data.

At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in an image 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 image can be perfectly recovered by exploiting the spatial correlation in natural image.

Algorithm 1 Procedure of Transformation

Input: An original image I and a secret key K.

Output: The encrypted image E(I).

1. Select a target image J having the same size as I from an image database.

2. Divide both I and J into several non-overlapping 4×4 blocks. Assuming that each image consists of N blocks, calculate the mean and SD of each block.

3. Classify the blocks with quantile of SDs and generate CITs for I and J respectively. Pair up blocks of I with blocks of J according the CITs.

4. For each block pair (Bi, Ti) ($1 \leq i \leq N$), compute the mean difference ui. Add ui to each pixel of Bi and then rotate the block into the optimal direction Oi (Oi ∈ {0o, 90o, 180o 270o}, which yields a transformed block T ′i.

5. In the target image J, replace each block Ti with the corresponding transformed block T ′ i for $1 \leq i \leq N$ and generate the transformed image J′.

6. Collect ui's and Oi's for all block pairs, and compress them together with the CIT of I. Encrypt the compressed sequence and the parameter Alfa by a standard encryption scheme such as AES with the key K.

7. Take the encrypted sequence as accessorial information (AI), and embed AI into the transformed image J′ with an RDH method such as the one in [7], and output the encrypted image E(I).

A. RDH in Encrypted Image

RIT generates an encrypted image E(I), which has the advantage of keeping a meaningful form of the image compared to traditional encryption methods. Therefore, it is free for the cloud server to employ any classical RDH on the encrypted image. Selecting what kind of RDH method depends on whether to keep the image quality or not. We simply adopt two RDH methods, one is a traditional RDH that keeps the quality of images and the other is a unified data embedding and scrambling method that may greatly degrades image structures for embedding large payload.

IV. CONCLUSION

This paper proposed a Scheme of RDH in encrypted images. After encrypting the original image with a stream cipher, some bits of MSB planes are selected and compressed to make room for the additional secret data. On the receiver side, all hidden data can be extracted with the embedding key only, and the original image approximately recovered with high quality using the encryption key only. When both the embedding and encryption keys are available to the receiver, the hidden data can be extracted completely and the original image recovered perfectly.

The embedding operations are performed to the encrypted data, the data-hider cannot access the contents of the original image, which ensures security of the contents in data hiding. As the embedding and recovery are protected by the encryption and embedding keys, an adversary is unable to break into the system without these keys.

Several interesting problems can be considered in the future, including how to improve the quality of the encrypted image and how to extend idea of RIT to audio and video.

REFERENCES

[1] "Reversible Data Hiding in Encrypted Images by Reversible Image Transformation.", Weiming Zhang, Hui Wang, Dongdong Hou, and Nenghai Yu, IEEE, August 2016.

[2] "Reversible Data Hiding in Encrypted Images with Distributed Source Encoding", Zhenxing Qian, Member, IEEE, and Xinpeng Zhang, Member, IEEE, APRIL 2016.

[3] "High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation", Xiaochun Cao, Senior Member, IEEE, Ling Du, Xingxing Wei, Dan Meng, Member, IEEE, and Xiaojie Guo, Member, IEEE, MAY 2016.

[4] "An Improved Reversible Data Hiding in Encrypted Images.", Shuang Yi, Yicong Zhou, IEEE, 2015.

[5] "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption." Kede Ma, Weiming Zhang, Xianfeng Zhao, Member, IEEE, Nenghai Yu, and Fenghua Li, IEEE, March 2013.

[6] "Separable Reversible Data Hiding in Encrypted Image." Xinpeng Zhang, IEEE, APRIL 2012.

[7] "Reversible Data Embedding Using a Difference Expansion". Jun Tian, IEEE, AUGUST 2003.