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. With the development of information technology the data are stored in the cloud and it need to be protect the privacy of data and management of the data at the same time. By these demands the reversible data hiding in encrypted images Reversible Data Hiding attracts more and more researcher's attention. Here propose a novel framework for Reversible Data hiding - Encrypted Image based on Reverse image Transformation. Here the content of the original image can be transform to the Content of another image. Then the transformed image, which looks like the target image, is used as the encrypted image, and sends to the cloud. Therefore, the cloud server can embed data into the encrypted image by using any RDH methods for plaintext images. RDH-EI is a client free scheme and the data embedding scheme is irrelevant with both process encryption and decryption. In the proposed method video framing technique is used to make video frames and these frames are embedded with a target image and store into the cloud and also improve the quality of encrypted image. This works improve the storage capacity cloud as well as the security of data.*

Keywords- Reversible Data Hiding, Reversible Image transformation, Cloud computing.

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]. Cloud computing is a emerging technology. In cloud computing a large pool of systems are connected in private or public networks. It is used to provide dynamically scalable infrastructure for application and storage of data and files. Cloud Providers offer services are classified into three categories they are Software as a Service(SaaS), Platform as a Service(Paas), Infrastructure as a Service(Iaas). Cloud Computing provides some benefits such as Reduced Cost, Increased Storage and Flexibility. 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. 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[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.

III. 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.

In the proposed system Secure Reversible Data Hiding Image Transformation using cloud storage where we can store videos in the cloud in a secured manner. The video frames are generated for the video which are uploaded by a user then a target image is chosen and each frame are encrypted and is transformed into the target image. These are all done in the user side. And these transformed frames are sending to the cloud. In the cloud side each frames are embedded with some data to make the images as watermarked images. The embedded additional data are removed on the download request by a user and the user get the encrypted images, by decrypting the images we get the original frames and by joining these frames the user get the original video. An authentication mechanism also used here for authenticating the user.

A. Block Pairing

To make the transformed image J′ look like target image J’, we hope, after transformation, each transformed block will have close mean and standard deviation (SD) with the target block. So we first compute the mean and SD of each block of I and J respectively. Let a block B be a set of pixels such that B = {p1, p2, … pn}, and then the mean and SD of this block is calculated as follows.

$$u=\frac{1}{n}\sum_{i=1}^{n} pi \qquad (1)$$

$$\sigma= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(pi-u)^2} \qquad (2)$$

When matching blocks between original image and target image, we hope two blocks with closest SDs to be a pair. the blocks of original image and target image are sorted in ascending order according to their SDs respectively, and then each original block is paired up with a corresponding target tile in turn according to the order. To recover the original image from the transformed image, the positions of the original blocks should be recorded and embedded into the transformed image with an RDH method.

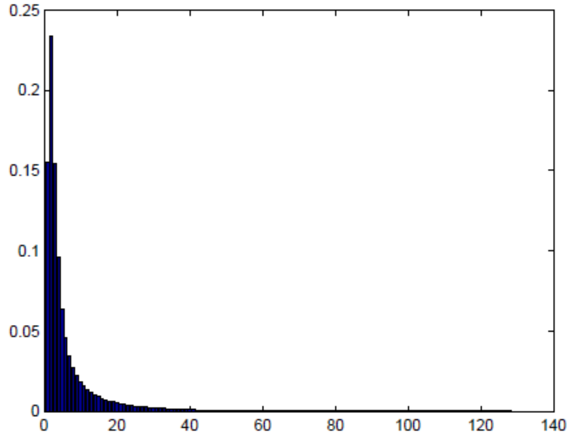

Figure 1. The distribution of SDs of 4 × 4 block for various sizes of natural images.

A. Block Transformation

By the block pairing method described above, in each pair (B, T), the two blocks have close SD values. Therefore, when transforming B towards T, we only need a mean shifting transformation that is reversible. Let the original block B = {p1, p2… pn}, and the Corresponding target block T = {p’1, p’2, ……. , p’n}. With Eq. (1), we calculate the means of B and T and denote them by uB and uT respectively.

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 %α 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) compute the mean difference Δui. Add Δui to each pixel of Bi and then rotate the block into the optimal direction Δi (Δɵ Ɛ {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 ≤ i ≤ N and generate the transformed image J’.
6. Collect Δui’s and Δi’s for all block pairs, and compress them together with the CIT of I. Encrypt the compressed sequence and the parameter α 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 ,and output the encrypted image E(I).

Algorithm 2 Procedure of Anti-transformation.

Input: The encrypted image E(I) and the key K.
Output: The original image I.

1. Extract ai and restore the transformed image j′ from e(i) with the rdh scheme.
2. decrypt ai by aes scheme with the key k, and then decompress the sequence to obtain cit of i, Δui, Δi (1 ≤ i ≤ n) and α.
3. divide j′ into non-overlapping n blocks with size of 4 × 4. calculate the sds of blocks, and then generate
1. the cit of j′ according to the %α quantile of sds.
4. according to the cits of j′ and i, rearrange the blocks of j′.
5. for each block t ′ i of j′ for 1 ≤ i ≤ n, rotate t ′ i in the anti-direction of Δi, and then subtract Δui from each pixel of t’i , and finally output the original image i.

IV. Experimental Results on RIT

In this section experimental results on the proposed RIT method are presented. 100 pairs of images are randomly chosen as our test images from the BossBase image database. Firstly all the images are preprocessed to get the same size of 1024 × 1024 pixels. Since in the RIT method the parameter α has an effect on the AI payload, we give the experiment to select a better α to improve the overall performance. The result is depicted in Fig. 2. The smaller the space occupied by AI is, the better the encrypted images visual quality will be. For α in the range of [0.05,0.95], the variation of AI payload seems to be not large. And it can be seen that when α is 0.75, the AI payload reaches the valley value. So in the following experiments, α is set 0.75.

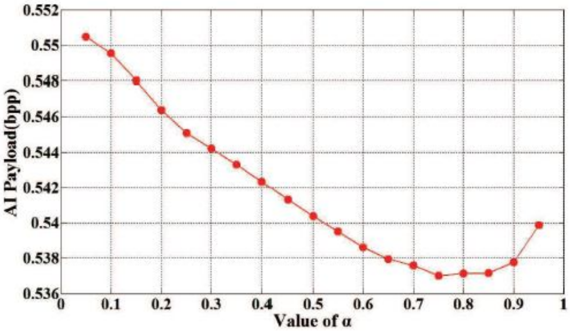

Figure 2. The effect of AI payload for different α values.

To illustrate the visual effect of the RIT method, experimental results of five pairs of test images labeled as A, B, C, D and E are given from Fig. 3 to Fig. 5. In Experiment A, we list the “decrypted images” with the right key and the wrong key respectively. Because the original image can be losslessly restored with the correct key, we did not list the “decrypted images” in the rest experiments. We also list the marked images with RDH in experiment D and E, which will be further discussed in the next section.

The encrypted images E(I) obtained by RIT look like mosaic images with their appearance similar to the target images. Since the difference between the encrypted image and the target image is small, such visual effect will meet the requirement of camouflage, which means that the original image content is totally covered by a target image content. Even if the attacker recognizes the camouflage, without the secret key K of AES, it is also unfeasible to decrypt the accessorial information that is necessary for restoring the original image. And thus the attacker only gets a meaningless image as shown in Fig. 5(e).

Table 1. Space Occupied by AI and PSNRS of the Encrypted Images

Experiment	A	B	C	D	E
AI(bpp)	0.523	0.499	0.521	0.554	0.508
PSNR(dB)	30.68	39.72	30.95	30.09	30.83

Experimental results of test images.

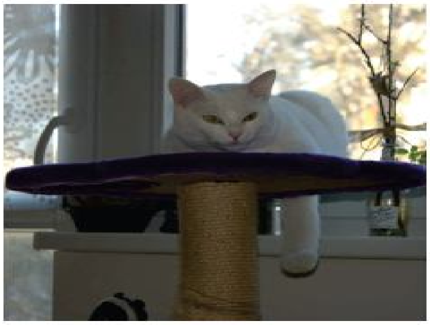

Figure 3. Original image

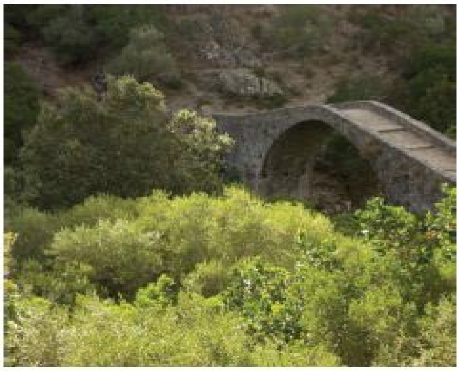

Figure 4. Target image

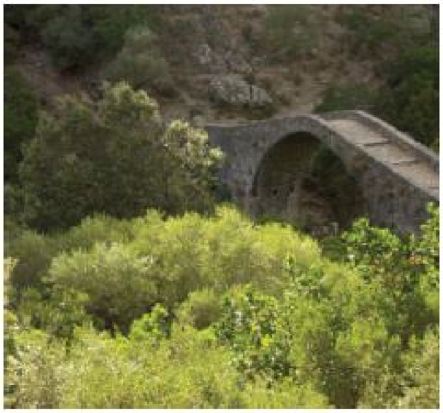

Figure 5. Encrypted image

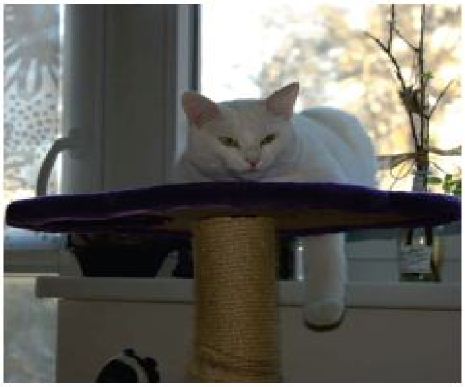

Figure 6. Decrypted image (right key)

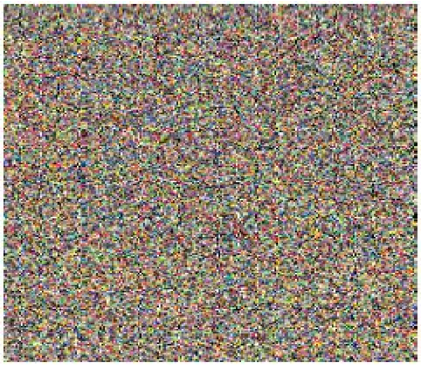

Figure 7. Decrypted image (wrong key)

Experimental results of test images.

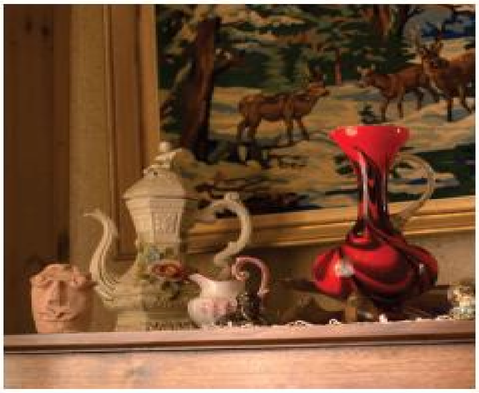

Figure 8. Original image

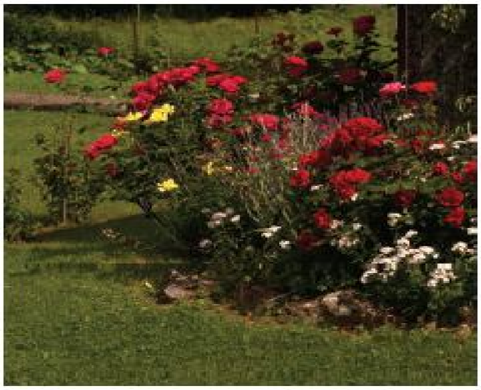

Figure 9. Target image

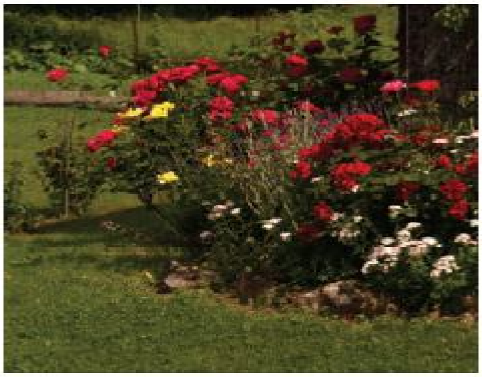

Figure 10. Encrypted image

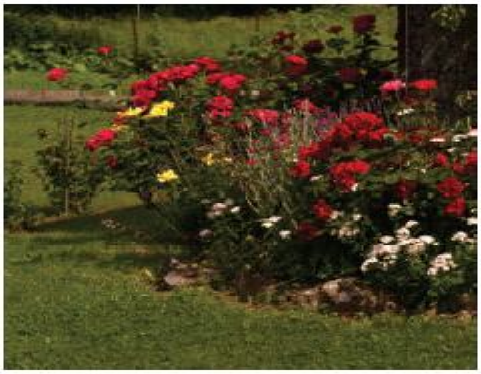

Figure 11. Marked image (0.1bpp)

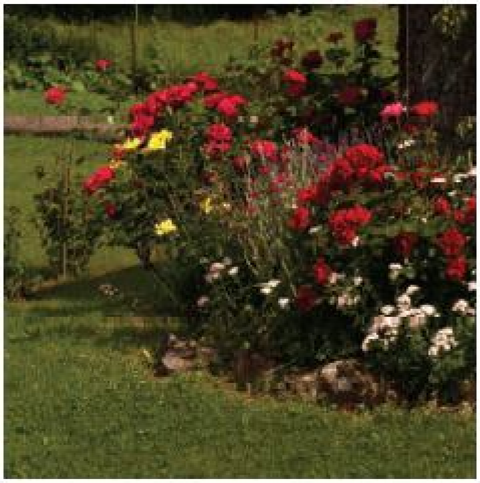

Figure 12. Marked image (0.5bpp)

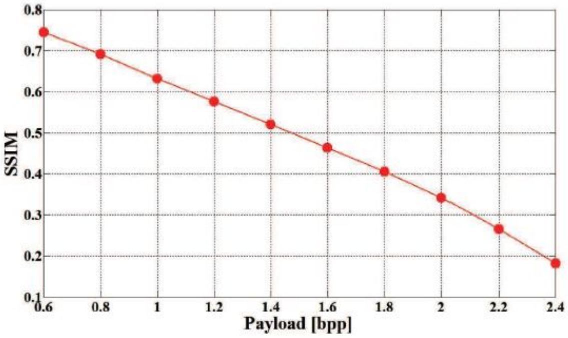

Figure 13. Average SSIM between encrypted images and marked images with different Embedding payloads for 100 pairs of test images by applying UES.

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. In this section 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. Fig. 5 shows that the average SSIM values for all test images are gradually decreasing with increasing payloads.

V. 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.