International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

Reversible Image Data Hiding Using Public Key Modulation over an Encrypted Domain

[1]Prithvi Muniraj, [2]Chandrakala H T

[1]M.Tech Student, Department of Computer Science and Engineering, BNM Institute of Technology, Bangalore, Karnataka, India
[2]Assistant Professor, Department of Computer Science and Engineering, BNM Institute of Technology, Bangalore, Karnataka, India

---

Abstract - *This paper proposes a reversible image data-hiding scheme over an encrypted domain to achieve data security. Data security is required to protect secret data from unauthorized access and data corruption. This scheme can be utilized in scenarios where data security plays a vital role for example military, medical science, banks and many more. The existing RIDH methods embed the data using a data-hiding key. This decrease the embedding capacity and extraction accuracy. To overcome these issues the proposed scheme emphasizes on embedding the data by eliminating the data-hiding key. Here, the embedding takes place on images encrypted using AES-CTR algorithm. Data embedding is performed using XOR operation via public key modulation mechanism on the sender end. On the receiver end, a 2-class SVM classifier is used to classify the embedded image into randomized and original image blocks. This results in carrying out data extraction and image recovery from the embedded image by identifying the blocks with the embedded data. When compared to the existing systems the proposed system proves to provide better embedding capacity and data extraction accuracy.*

Key Words: **Reversible image data hiding, data embedding, SVM classifier, data extraction and image recovery**

1. INTRODUCTION

Data hiding is a method of enclosing data into cover objects like images, signals, audio, video and text. Image data hiding is defined as embedding data into images. A special technique reversible image data hiding (RIDH) is an image data hiding technique, in which the cover image is reconstructed to perfection upon extraction of the secret message. This technique overcomes the disadvantages of irreversibility, where in the cover image is distorted when recovered. This technique is most particularly suitable in scenarios like military, secure remote sensing, cloud computing, medical image sharing etc. The existing method mainly utilized lossless compression algorithms in which some of the image features are compressed to make room for embedding the data. However, data-embedding capacity is decreased and distortion of the recovered image is increased. To overcome these disadvantages histogram shifting method is introduced in which histogram peak and zero points were shifted to vacate space for data embedding. The new technique in the field of RIDH to embed the data within the cover image is difference expansion. Here, the adjacent or neighbor pixels are considered to calculate the pixel difference. This pixel difference is used to create a LSB plane into which the data can be embedded. The method increased the embedding capacity and accuracy when compared to the state of art algorithms developed.

When the data hiding process takes place in the existing methods, to maintain the embedding security a data-hiding key has to be provided. To maintain the overall security of the system; it is necessary for message embedding to take place in an encrypted domain due to the presence of data-hiding key. Encrypted domain is required to make sure the data-hiding key is secure from attackers. For instance in the cloud computing domain to maintain the security and privacy of the data RIDH technique can be used. Data embedding in this technique takes places using a data-hiding key. This imposes a need to build a Key management system in multiparty and insecure environment of the cloud. KMS is required to manage the key operations like creation, deletion, activation, deactivation etc. Therefore, building a KMS is very costly, requires more memory space, extra security and proves to be a disadvantage of the existing methods. Hence, a system that does not require data-hiding key to maintain the embedding security has to be developed. In addition to this, it is appreciated to use a simple embedding algorithm as the systems are usually constrained by limited computing capabilities and/or power.

In the proposed approach, reversible image data hiding scheme is developed considering the above disadvantages of the existing methods. This approach mainly focuses on data embedding without using a data-hiding key. It falls under the category of non-separable reversible image data hiding. The proposed scheme on the sender end uses AES-CTR algorithm to encrypt the image and divides the image into blocks. Public key modulation mechanism is used to embed the data into the blocks of the encrypted image. This mechanism is developed mainly

to overcome the disadvantage of building a KMS. On the receiver end the extraction process takes place firstly by classifying the embedded image into original blocks and encrypted blocks using a 2-class SVM classifier. Later, the data extraction and image recovery is accomplished simultaneously. Image is recovered using AES-CTR decryption algorithm. The proposed scheme provides better embedding capacity and recovers the image and data to perfection. Substantial experimental results are carried out on 50 test images to validate the performance of the proposed scheme.

2. RELATED WORK

Some of the recent attempts were made to develop Reversible data hiding algorithms to embed data into encrypted images. Here, a novel reversible data hiding technique, data embedding takes place over encrypted image divided into blocks of size s*s. These pixels of each block are then classified into two types S_0 and S_1 based on the data-hiding key. When the data to be hidden is binary digit 0 then 3 LSB's of encrypted pixel S_0 are flipped. If the additional bit is binary digit 1 then 3 LSB's of S_1 are flipped. The image decryption is accomplished using the secret encryption key and data is extracted using the data-hiding key [1].

This is an improvised work of [1]. The original work focuses on embedding the data into specific pixels determined using the data-hiding key. The main drawback of this is not all the pixels are utilized and pixel correlation in the border of neighboring blocks is not considered. In addition, the block smoothness is also not calculated. This decreases the data extraction accuracy and the image recovered is distorted. To overcome these issues a end-match scheme is proposed to reduce the errors in the extraction process. The block smoothness is also calculated using the absolute difference of the neighboring pixels. More the summation of the difference more complex the image blocks is. Therefore, the block smoothness is estimated by calculating the summation of the vertical absolute differences and horizontal absolute differences of pixels in image blocks [2].

This work focuses mainly on two dimensional histogram difference and a reversible data hiding scheme is proposed which uses the difference pair mapping technique (DPM). Firstly, each pixel-pair and its context are examined; a sequence consisting of pairs of difference values are computed. Then, a two-dimensional difference-histogram is created by counting the frequency of the resulting difference-pairs. The data embedding is finally achieved using the specifically designed DPM. Here, the DPM is an injective mapping defined on difference-pairs. It is an enhanced version of linear expansion and histogram shifting techniques used to embed the data in current histogram-based RDH methods. It method provides better embedding capacity and less distortions [3].

In this approach data-embedding is carried out using the pixel difference. The image is divided into blocks based on the pixel difference and histograms are generated for each blocks. The data embedding is carried out using a pixel difference of single pixel value. The number of bits to embedded per block is determined is calculating the peak values of the histograms. If the image contains blocks with uniform pixels then embedding is not possible this reduces the embedding capacity. To overcome this problem the difference between the adjacent pixels are considered. The distortion of the recovered image is less compared to the above methods [4].

In a real-time situation it sometimes required to implement reversible data hiding on color images. The encryption standards used to encrypt the color images requires large amount of time to encrypt and decrypt the image. To overcome this disadvantage a chaotic encryption is used to secure cover image. The chaotic encryption encrypts selective pixels and introduces confusion and diffusion in the encryption process. Since the method, uses reverse room before encryption method (RRBE) recovery of the data and image can be implemented losslessly. This method achieves real reversibility when compared to other methods that vacate room before encryption [5].

A lossless, reversible, and combined data hiding scheme for cipher text images encrypted by public-key cryptosystems with homomorphic and probabilistic features is proposed. In lossless scheme, the secret data is embedded by replacing the encrypted pixels with new values. The data gets embedded into LSB planes of the encrypted pixels using multilayer wet paper coding. The main advantage of the lossless scheme is the hidden data can be extracted directly and it does not affect the reconstruction of the original cover image. In reversible scheme, to hide the data the histograms of the image are compressed using a preprocessing method. Then the image is encrypted to avoid pixel saturation for data embedding. Even though, a slight distortion is produced, the embedded data can be extracted to perfection. Due to co-operation between the lossless and reversible schemes, the data-embedding operations in the two manners can be performed in an encrypted image. With the combined technique, a receiver may extract a part of embedded data before decryption, and extract another part of embedded data and recover the original plaintext image after decryption [6].

In this approach the images are encrypted before embedding by using stream cipher and josephus traversal. A new approach known has block histogram shifting is developed which uses self-hidden histogram peaks to perform Reversible data hiding. The main advantage of this work is when the receiver obtains the secret encryption key only the image can be decrypted. The data-hiding key is required to extract the data. If both the keys

are available, the extraction of the image and data takes place without errors. This work provides higher embedding payloads and better quality of the decrypted images [7].

3. PROPOSED APPROACH

The proposed approach concentrates on embedding the data into the encrypted image to attain data security. The system architecture of the proposed work is diagrammatically represented in Fig-1. On the sender end image encryption and data embedding takes place. The embedded image transmitted to the receiver using a public channel like wifi, LAN etc. This public channel can be coded using socket programming concept in java language. On the receiver end, the focus is on data extraction and image recovery.

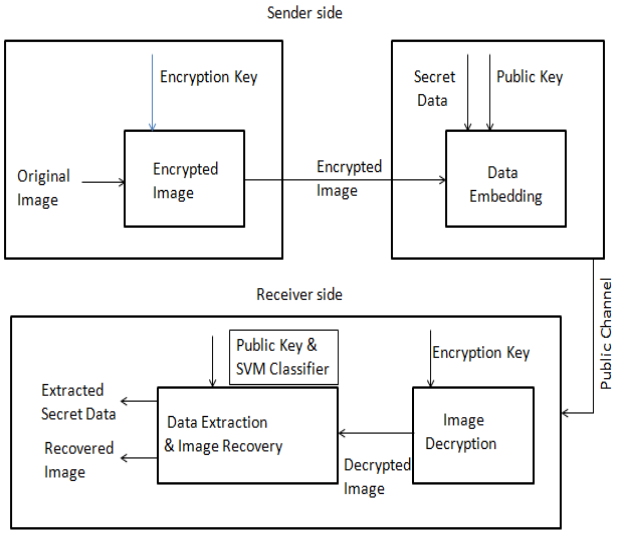


Fig-1: System architecture

Firstly, in the Reversible image data-hiding scheme the cover image that acts as the media to embed the data must be encrypted. This is accomplished using the standard stream cipher algorithm the AES-CTR. The main advantage of using the stream cipher algorithm over the existing algorithm used in the existing method as XOR is:

- AES-CTR is block cipher mode algorithms which converts block cipher to stream cipher.
- AES-CTR provides the advantage of inducing confusion and diffusion in the encrypted image produced.
- It overcomes the linear cryptanalysis attack in which the action of cipher values produced can be predicted.
- It contains non-linear elements which makes to difficult for the attacker to decode the image

Hence, due the implementation of the AES-CTR algorithm it can be told the RIDH technique takes place over an encrypted domain.

3.1 Image Encryption and Data embedding

The encryption of the image on the sender end can be achieved using the following pseudo-code.

Begin

Input: Counter and key

1. Initializing counterh
   i. Counterh = 16 bit counter
   ii. keyh = randomly generated 16-bit key
   iii. stateh = Counter XOR key
2. Creation of state matrix from stateh
   i. state = reshape(in, 4, 4);
3. Carrying out AddRoundKey operation
   i. state = bitxor(state, (s.keyexp(1:4, :))');
4. For i = 1:(s.rounds - 1)
   i. state = s.s_box(state + 1);
   ii. state = shift_rows(state, 0);
   iii. state = mix_columns(state, s);
   iv. state = bitxor(state, (s.keyexp((1:4) + 4*i, :))');
   b. end for
5. The encrypted (counter, key) = K is produced.
6. Encrypted image [[I]] = I XOR K

End

Once the image is encrypted the image is divided into non-overlapping blocks of size M*N and with the block index i. Each block can be embedded with n bits of data and embedding capacity can be calculated using n.B where B represents the number of blocks in the cover image. To carry out data embedding it is necessary to develop public key X_0, X_1, X_2 ..... X_{n-1}. The total number of public keys to be generated are $V= 2^n$ where n represents the number of bits that can be embedded into the blocks of the image. The public keys are generated using a random number generator and are stored in matrix. The main reasons for developing the public key matrix are these matrices are in built into the sender and the receiver system. Hence, this eliminates the need to transmit the keys across the network.

The steps required data embedding with the aid of the public keys as well as the secret message that is user entered and converted to binary format are:

Step 1: Initialize the block index = i

Step 2: Convert the user-entered data into binary bits.

Step 3: Fetch the n-bits of data to be embedded to the block denoted by Y_i

Step 4: Find the required number of public keys from the public key matrix X_d

Step 5: Equation (1) is used to carry out the embedding process that is implemented using the XOR operation

$$[[I]]_i^y = [[I]]_i \; XOR \; X_{[Y_i]d} \qquad (1)$$

Step 6: Increment the value of index to i=i+1 and repeat step 2-5 until all the bits are embedded

The above steps prove that data embedding is carried out without using a data-hiding key. The high-level of embedding security can still be obtained has the secret encryption key is encrypted using the standard AES algorithm. The elimination of the data-hiding key is not a unique feature of the proposed work as almost all the non-separable RIDH scheme use the same. The removal of the data hiding key can lead to decrease in computational cost and the risk involved in building and KMS , which has been proved to be very challenging in the multiparty environment [8]. Even though the possibility of removing the data hiding key holds for all non-separable RIDH schemes over encrypted domain in theory, it has never been practically implemented in the existing work. It can be seen that all the existing system with respect to the RIDH scheme are implemented using the data-hiding key. Once the embedded image is obtained, the socket programming using the IPaddress and socket number is coded to transfer the image to the receiver.

3.2 Support vector machine classification for differentiating encrypted and non-encrypted blocks

The feature vector designed to discriminate the original and the encrypted blocks are entropy indicator and standard deviation it is represented as $\rho = (H, \sigma)$'. H represents the entropy indicator and σ represents standard deviation.

The pixels in the encrypted blocks are more uniformly distributed when compared to the pixels of the original blocks. The entropy indicator is used to measure the randomness of the pixels in the blocks containing the embedded data. Since the feature extraction is taking place on the blocks of the image, it is always necessary to be cautious. When the block size decreases the classification accuracy decreases i.e. when M=N=8 the feature extraction can only take place on 64 pixels but it requires 0-255 pixels. To overcome this problem first the quantization step is performed in accordance with the block size. Later, on the quantized pixel samples entropy indicator is applied. The uniform scalar quantization applied on each of block is as follows:

$$F = \frac{MN.f}{256} \qquad (2)$$

Where, F and f denote the quantized and original pixel values respectively. The Entropy indicator H can be calculated on the quantized samples as follows

$$H = -\sum_{j=0}^{MN-1} p(j) \log p(j) \qquad (3)$$

p(j) represents the empirical probability and MN is the block size of the image. Only the entropy indicator is not enough to calculate the features of the blocks. Therefore, standard deviation is calculated on the quantized samples as follows

$$\sigma = \sqrt{\frac{1}{MN} \sum_j (f(j) - \mu)^2} \qquad (4)$$

Here μ represents the mean over all the pixels of the block and is calculated as

$$\mu = \frac{1}{MN} \sum_j f(j) \qquad (5)$$

f(j) represents the j^{th} pixel of the block. By including the standard deviation the dispersiveness and the denseness of the blocks can be obtained accurately. The SVM classifier is trained using RBF-Gaussian kernel and this classifier can also be called as offline trained SVM classifier. The blocks are classified into 2 classes namely:

Class 0 – corresponds to original blocks

Class 1 – corresponds to encrypted blocks

3.3 Image recovery and Data extraction

Once the embedded image is received on the receiver end using the socket and IPaddress, the image is decrypted using the AES-CTR algorithm and classified using the SVM classifier once the image blocks with embedded data are recognized the data extraction can takes place.

The pseudo-code for AES-CTR algorithm is as follows

Begin

1. Load the encrypted (counter, key) = K from the sender end
2. Decrypt the embedded image using $I^y = [[I]]_i^y \; XOR \; K$

End

Once I^y = embedded image is obtained, apply the SVM classifier to classify the blocks of the image. The data extraction takes place by

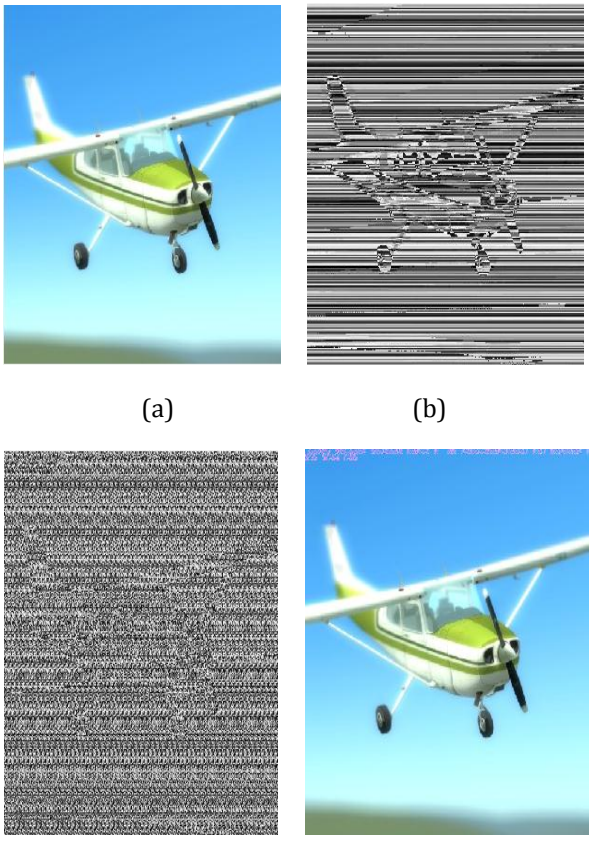
$$Y_i = I^y XOR\ X_d \qquad (6)$$

Here Y_i is the extracted data and X_d represents the public keys in the matrix stored.

NOTE: The proposed RIDH scheme over encrypted domain may also be extended to handle compressed and encrypted images, namely, embed watermark into the compressed and encrypted bit stream.

4. EXPERIMENTAL RESULTS

In this section, the embedding performance of the proposed RIDH scheme is experimentally evaluated. Image datasets are collected from decsai.ugr.es that consists of standard 512*512 grayscale test images and homepages.cae.wisc.edu contains 512*512 color images. As mentioned in the section III, the standardized encryption algorithm AES-CTR is used to encrypt and decrypt the images. The Fig-2 shows the results of the proposed approach.



(a) (b)

(c) (d)

Fig-2: Results of RIDH scheme (a) original image (b) encrypted image (c) embedded image (d) recovered image.

The proposed approach is compared with two existing state-of-art algorithms [18] – [19]. The comparison is conducted based on three parameters the data extraction accuracy (tau), image extraction accuracy and embedding capacity for different block sizes. The tau is calculated using the formula

$$\tau = \left(\frac{number\ of\ recover\_wrd}{number\ of\ WORD}\right) * 100 \qquad (7)$$

Where recover_wrd = number of correctly extracted bits and WORD = number of embedded bits. The results are averaged over all the blocks in the 50 test images. The image recovery accuracy is calculated is using the formula

$$Accuracy = \left(\frac{1-k}{h*w}\right) * 100 \qquad (8)$$

Where, k = number of error blocks identified using the 2-class SVM classifier by comparing the encrypted image and original image, h = height of the image and w = width of the image.

The number of bits to be embedded into blocks is n=2 i.e., 2 bits of data can embedded into the blocks of the image. The tested block sizes are 8*8, 7*7, 6*7 and 5*4. The performance analysis based on data extraction accuracy and image extraction accuracy is graphically represented in Chart-1.

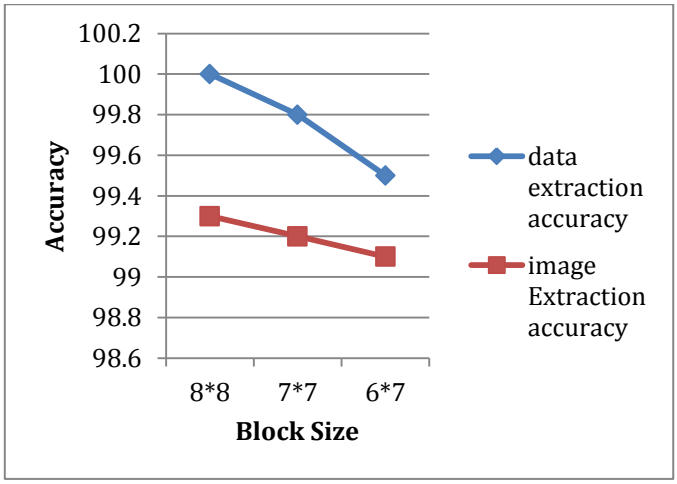


Chart-1: Performance analysis of data and image extraction accuracy

When the number of bits to be embedded n increases the accuracy of the data extracted decreases. Furthermore, it is investigated that the effect brought by increasing n, i.e., embed more bits into one single block.

Obviously, the number of public keys Xj's exponentially increases as n is larger. This will enlarge the complexity of data extraction needed to examine the entire $S = 2^n$ decoding candidates. The Table-1 and Chart-2 represents

comparative analysis carried out with the existing methods.

Table-1: Comparative analysis between proposed system and existing systems.

	Proposed system		RDH in encrypted images		RDH using side match	
Block Size	Capacity	Accuracy	Capacity	Accuracy	Capacity	Accuracy
8*8	8192 bits	100%	4096 bits	89.44%	4096 bits	92.04%
7*7	10,658 bits	99.8%	5329 bits	87.20%	5329 bits	90.65%
6*7	12,410 bits	99.5%	6205 bits	84.19%	6205 bits	88.88%

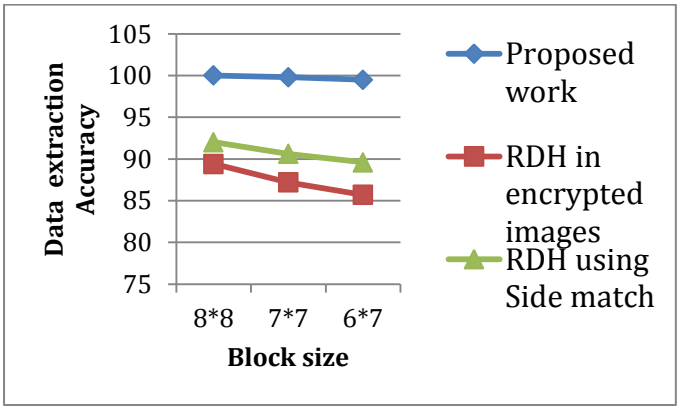


Chart-2: Comparative analysis with existing systems

The proposed approach is implemented on two systems, which act as the sender and receiver respectively. The work is implemented in MATLAB built-in tic and toc functions in a personal PC with Intel i7@3.40-GHz CPU and 32-GB RAM and Netbeans IDE. Netbeans IDE is used to code for socket program required to transfer the embedded image from sender to receiver. Note that the joint image decryption and data extraction of different blocks are largely independent. It can also be pointed out that the complexity of performing the joint image decryption and data extraction may not be crucial in many applications, e.g., secure remote sensing, where the recipient has abundant computing resources.

5. CONCLUSION

In this paper, a secure Reversible image data-hiding scheme is designed over an encrypted domain. This scheme can be used in various scenarios like military, medical data sharing, authentication and many more. A powerful stream cipher algorithm AES-CTR is used to encrypt and decrypt the images, which forms the encrypted domain. To overcome the disadvantages of the existing methods a public key modulation mechanism is used to embed the data without accessing the secret encryption key. It also eliminates the need of using an extra data-hiding key. Data embedding is carried out via simple XOR operations at the sender end. At the receiver end, a powerful two-class SVM classifier is used to discriminate randomized and original image patches. This enables simultaneous decoding of the embedded message and the cover image to perfection. Experimentation was performed to validate the embedding performance and embedding capacity of the proposed RIDH method over encrypted domain to obtain good results.

REFERENCES

[1] Xinpeng Zhang, "RDH in encrypted images", IEEE signal processing letters, vol.18, no.4, pp. 1070-9908 IEEE April2011.

[2] Wien Hong, Tung-shou Chen, and Han-Yan Wu, "An improved reversible data hiding in encrypted images using end match," IEEE Signal Process. Lett., vol. 19, no. 4, pp. 199–202, Apr. 2012.

[3] Xiaolong Li, Weiming Zhang, Xinlu Gui, and Bin Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," IEEE Trans. Inf. Forensics Security, vol. 8, no. 7, pp. 1091–1100, Jul. 2013.

[4] Lincy Rachel Mathews, Arathy C. Haran V., "Histogram Shifting Based Reversible Data Hiding Using Block Division and Pixel Differences", 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) 978-1-4799-4190-2/14 © IEEE December 2014 .

[5] Besteena K J, Philumon Joseph, "RDH in Selectively Encrypted RGB Images by Reserving Room in Advance", 2014 First International Conference on Computational Systems and Communications (ICCSC) 978-1-4799-6013-2/14 2014 IEEE 17-18 December 2014.

[6] Xinpeng Zhang, Member, IEEE, Jing Long, Zichi Wang, and Hang Cheng, "Lossless and Reversible Data Hiding in Encrypted Images with Public-Key Cryptography", IEEE transactions on circuits and systems for video technology, vol. 26, no. 9, September 2016.

[7] Zhaoxia Yin, Andrew Abel, Xinpeng Zhang and Bin Luo, "Reversible data hiding in encrypted image based on block histogram shifting," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, 2016, pp. 2129-2133.

[8] M. Chandramouli, R. Iorga, and S. Chokhani, "Publication citation: Cryptographic key management issues & challenges in cloud services," US Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 7956, 2013, pp. 1–31.