A Homomorphic Method for Sharing Secret Images

Naveed Islam, William Puech, Robert Brouzet

A Homomorphic Method for Sharing Secret Images

Naveed Islam[1], William Puech[1], and Robert Brouzet[2]

[1] LIRMM Laboratory, UMR 5506 CNRS, University of Montpellier II
34392 MONTPELLIER FRANCE
[2] I3M Laboratory, UMR 5149 CNRS, University of Montpellier II
34392 MONTPELLIER FRANCE

Abstract. In this paper, we present a new method for sharing images between two parties exploiting homomorphic property of public key cryptosystem. With our method, we show that it is possible to multiply two encrypted images, to decrypt the resulted image and after to extract and reconstruct one of the two original images if the second original image is available. Indeed, extraction and reconstruction of original image at the receiving end is done with the help of carrier image. Experimental results and security analysis show the effectiveness of the proposed scheme.

Cryptosystem, Homomorphism, Image encryption.

1 Introduction

With the development of new communication technologies, Internet transfer of visual data (images, videos or 3D objects) for different types of multimedia applications has grown exponentially. However, digital communication is increasingly vulnerable to malicious interventions or monitoring like hacking or eavesdropping. The security of these sensitive visual data in applications like safe storage, authentication, copyright protection, remote military image communication or confidential video conferencing require new strategies for secure transmission over insecure channel. There are two common techniques used for secure transmission of data namely cryptography and watermarking. Cryptography ensures the security by scrambling the message using some secret keys [9]. Homomorphic cryptosystems are special type of cryptosystems which preserve group operations performed on ciphertexts. A homomorphic cryptosystem has the property that when any specific algebraic operation is performed on the data input before encryption, the resulting encryption is same as if an algebraic operation is performed on the data input after encryption [8]. Homomorphic property of public key cryptosystems has been employed in various data security protocols like electronic voting system, bidding protocols, cashing systems and asymmetric finger printing of images [4]. The use of carrier image for the encryption of image has been presented in [6] using private key cryptosystem in frequency domain. For the authentication of images, copyright protection, watermarking techniques are

used, these watermarking techniques along with cryptographic technique gives enough level of security [7]. In this paper, we exploit the multiplicative homomorphic property of RSA cryptosystem for sharing secret images using carrier image for both transfer and extraction of original image.

This paper is organized as follows. In Section 2, we first give a brief introduction of cryptographic techniques focusing on asymmetric encryption of RSA with special reference to its homomorphic property and then we explain how to apply it to an image. The proposed algorithm is detailed in Section 3 and experimental results along with security analysis of the proposed scheme are studied in Section 4. Finally, Section 5 gives summary and concluding remarks.

2 Previous works

Extra storage capacities and special computation is required for visual data types such as images, videos or 3D objects, due to the large amount of data. Nowadays cryptographic techniques for image security are widely used for secure transfer. In image domain, there may be full encryption or selective encryption of the image depending on the application. Since many applications require real time performances, partial encryption is mostly used [10]. Cryptographic techniques can be divided into symmetric encryption (with secret keys) and asymmetric encryption (with private and public keys).

In symmetric cryptosystems, the same key is used for encryption and decryption. Symmetric key cryptosystems are usually very fast and easy to use. Since same key is used for encryption and decryption, the key needs to be secure and must be shared between emitter and receiver.

2.1 Asymmetric encryption

In asymmetric cryptosystem, two different keys are necessary: the public and the private keys. With the receiver public key, the sender encrypt the message and send it to the receiver who decrypt the message with his private key. Some known algorithms are RSA, El Gamal and Paillier cryptosystems [9, 3, 5]. RSA and El Gamal are public-key cryptosystems that support the homomorphic operation of multiplication modulo n and Paillier cryptosystem support homomorphic addition and subtraction of encrypted messages.

RSA is a well known asymmetric cryptosystem, developed in 1978. The general procedure consists of selecting two large prime numbers p and q, calculating their product $n = p \times q$ and selecting an integer e, which is relative prime to $\Phi(n)$ and with $1 < e < \Phi(n)$, where $\Phi(n)$ is the Euler's function. We need to calculate d, the inverse of e with $d \equiv e^{-1} mod\, \Phi(n)$. The public key is composed of the couple (e, n) and the private key of the couple (d, n). For the encryption, the plaintext M is partitioned into blocks m_i such that $m_i < n$ and for each plaintext m_i we get a ciphertext c_i:

$$c_i = m_i^e \; mod \; n. \tag{1}$$

For the decryption, with the ciphertext c_i we can obtain the original plaintext m_i by the equation:

$$m_i = c_i^d \ mod \ n. \tag{2}$$

Example: assume primes p and q are given as $p = 7$,$q = 17$ therefore $n = p \times q = 7 \times 17 = 119$, let $e = 5$, which follows that $gcd(\Phi(p * q), 5) = 1$ and for $e = 5$, we found $d \equiv e^{-1} \ mod \ \Phi(n) = 77$. Let the input plain texts be $m_1 = 22$ and $m_2 = 19$. Therefore the encryption of m_1 is given as: $c_1 = 22^5 mod \ 119 = 99$ and the encryption of m_2 is given as: $c_2 = 19^5 \ mod \ 119 = 66$.

2.2 Multiplicative homomorphism

Most of the asymmetric cryptosystems follow either additive homomorphism or multiplicative homomorphism. An encryption algorithm $E()$ is said to be homomorphic if it obeys the following condition [2]:

$$E(x \oplus y) = E(x) \otimes E(y), \tag{3}$$

where $\oplus$ and $\otimes$ can be addition, subtraction or multiplication and not necessary the same between the plaintexts and the ciphertexts. But usually the former operation is either addition or multiplication or exclusive or while the latter is multiplication.

The encryption algorithm RSA follows multiplicative homomorphism:

$$E(m_1) \times E(m_2) = E(m_1 \times m_2). \tag{4}$$

Example: with the values of the example presented in Section (2.1) we have $c_1 \times c_2 = 99 \times 66 \ mod \ 119 = 108$. Multiplying the two plaintexts will give a third text m_3 given as: $m_3 = m_1 \times m_2 = 22 \times 19 \ mod \ 119 = 61$. The encryption of m_3 is given by: $c_3 = 61^5 \ mod \ 119 = 108$ which equals to the multiplication of two ciphertexts. Hence RSA support homomorphic operation of multiplication modulo n, presented in equation (4).

2.3 Image encryption

Extreme care must be taken while calculating the values of the keys because the security of encrypted image depends on the size and the value of the public key and small or bad keys can produce encrypted images which contain information of the original images [1]. An effective way for image security using asymmetric cryptographic techniques is block-based image encryption. In block-based image encryption schemes the block size is selected according to the size of the key, so that encrypted data provide sufficient level of security in shape of key size and no extra payload in shape of increase in image size appears. Also the creation of block and then encryption should be made in such away that the ciphered image does not reveals any structural information about the data in the image. For the proposed method, the image is transformed into a coefficient image, where each

coefficient has size equal to the size of the block in the original image and the block size depends on the key size being selected for encryption and decryption. If length of the encryption key is γ bits then the number of pixels in the block is given by:

$$b = \lceil \gamma / k \rceil \,, \tag{5}$$

where k is the number of bits of a single pixel. Let an image of size $M \times N$ pixels $p(i)$, where $0 \leq i < M \times N$, the construction of the coefficient values from the original image pixels $p(i)$ is given as:

$$B(i) = \sum_{j=0}^{b-1} p(i * b + j) \times 2^{kj}, \tag{6}$$

where $0 \leq i < \lceil M \times N / n \rceil$ for the coefficient image.

For RSA cryptosystem, to be applied on each coefficient, let $B(i)$ be the ith constructed coefficient of an image, then the encryption of $B(i)$ is given by:

$$B'(i) = E_k(B(i)) = B(i)^e \; mod \; n, \tag{7}$$

where $B(i)$ and $B'(i)$ are coded on γ bits of each coefficient. After decryption of $B(i)$, the decomposition of the transformed coefficients to get the original pixels is given by:

$$\begin{cases} \text{if } j = 0 \\ p(i \times b + j) = B(i) \; mod \; 2^k \\ \text{else} \\ p(i \times b + j) = \Big(B(i) \; mod \; 2^{k(j+1)} - \sum_{l=0}^{j-1} p(l) \Big) / 2^{kj} \end{cases}$$

3 Proposed homomorphic based method

3.1 Standard protocol for image transmission

The standard protocol for secure image or message transfer is based on the security of the keys. In standard procedure, if a user P1 wants to send image M1 to user P2, he will first encrypt the image with the public key of the receiver i.e. P2. This encrypted image will be then transmitted to the user P2 over unsecured transmission channel. At the receiving end, in order to read the image, the user P2 will decrypt the image with his private key, as shown in Fig. 1.

For authentication, the protocol is a little bit changed, the sender must first encrypt the sending image with his private key and then again encrypt with the receiver public key, the first encryption allow him to sign the sending message. Similarly the receiver first decrypt the message with his private key and then for authentication he will use the public key of the sender for decryption, as illustrated in the Fig. 2. But here two keys are required by each user and also the processing time for encrypting and decrypting increases.

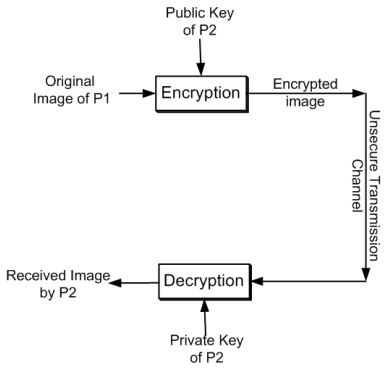


Fig. 1. Standard way for image transmission.

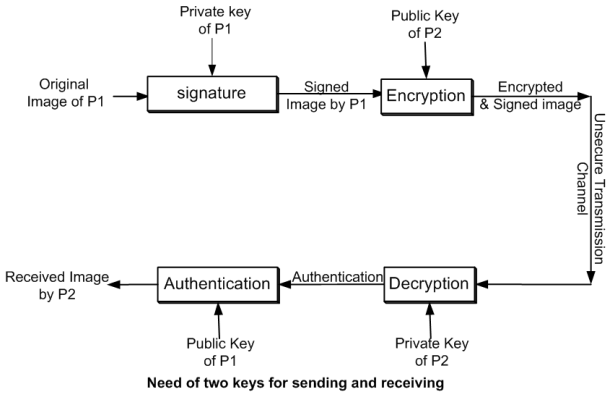


Fig. 2. Standard way for image transmission along with authentication.

3.2 Overview of proposed method

The purpose of our scheme is to securely transfer and to share a secret image between two persons. Even if an intruder gets a copy of the protected transmitted image he can not be able to extract the original image. A block diagram of encryption step of proposed technique is given in Fig. 3.

Each user takes an image of same size and transform it into a coefficient image using equation (6), where each coefficient represents the total number of pixels in a single block, we then apply asymmetric algorithm of RSA on each coefficient of the transformed image. Note that the same key is used for encryption process separately for both images. After the two images have been encrypted, we take modulo multiplication of the two encrypted images to get a third encrypted image. Because of the homomorphic property of RSA, this third encrypted image must be the same if we had first multiplied the two original images to get a third image and then applying RSA algorithm. The third encrypted image or its decrypted version can be transferred over any insecure channel. Since the third image contains components of both first and second original images, one

can extract any one of the two original images if other image is available. At the receiving end, as a user has one of the original images and he received the third image, he can extract the second original image with the help of his own image. This extracted image contains noise elements because some encrypted pixels can give multiple solutions during the extraction. So, we apply a reconstruction algorithm in order to remove the maximum of the noise pixels and get better pixels. Fig. 4 shows the block diagram of the proposed method for decryption.

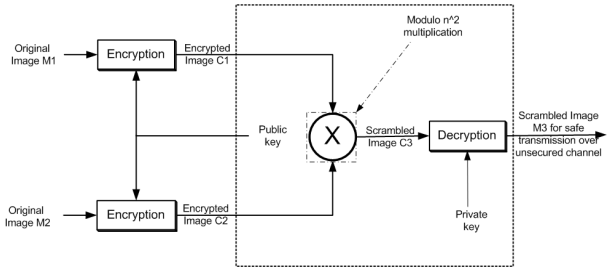


Fig. 3. Overview of encryption step.

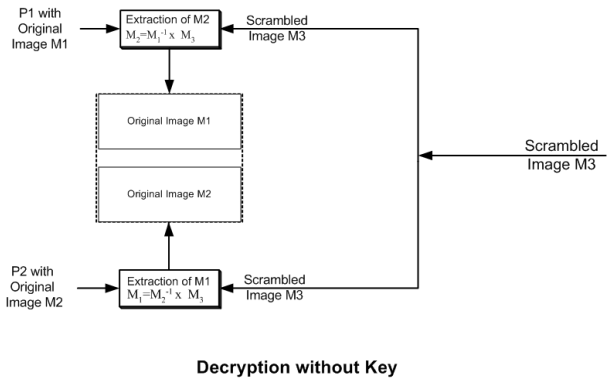


Fig. 4. Decryption of scrambled image without use of public or private keys.

3.3 Encryption step

For each block of the two original images M_1 and M_2 we apply the RSA encryption as described in equation (7). The image M_1 is considered to be available at both ends. But before encryption, some preprocessing must be done due to limitation on encryption algorithm and image data size.

After the encryption of the two original images M_1 and M_2 we get the two encrypted images C_1 and C_2 as illustrated in Fig. 3. We can then scramble these

two encrypted images by applying a modulo multiplication between them. Since each block of the two encrypted images C_1 and C_2 has value between 0 and $2^\gamma - 1$, after the multiplication of the two encrypted images we must apply the modulo operation to get scrambled pixels of C_3 encodable on γ bits:

$$B'_3(i) = B'_1(i) \times B'_2(i) \; mod \; n. \tag{8}$$

This encrypted image C_3 can be decrypted with the private key to produce M_3. This M_3 is our intended image to be transferred by the sender to the receiver through insecure channel.

3.4 Extraction and reconstruction

The block diagram of the proposed method for extraction and reconstruction is shown in Fig. 5. At the receiving end, for example user P_1 has M_1 and receives M_3 and then wants to extract M_2. Due to the multiplicative homomorphic property of RSA, from the equation (8), we have also:

$$B_3(i) = B_1(i) \times B_2(i) \; mod \; n. \tag{9}$$

We can do inverse modulo operation of equation (9), which gives single values for the coefficients $B_1(i)$ of M_1 which are relative prime to n and multiple solutions for the coefficients $B_1(i)$ which are non relative primes to n. For these particular cases the reconstruction step consists in choosing the best value among the multiple solutions for particular blocks in order to try to reconstruct an image the nearest to the original image M_2.

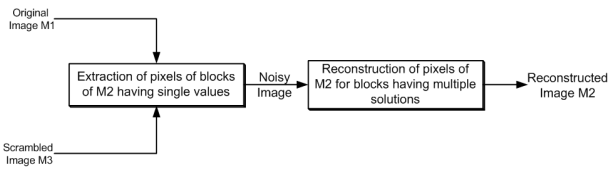


Fig. 5. Extraction and reconstruction of image M2 having pixels of image M1.

In order to explain the principles that make the extraction of the second image M_2 possible, let us consider that p and q are primes such that $p < q$, and $n = p \times q$. Let $B_1(i)$, $B_2(i)$ and $B_3(i)$ three integers between 0 and $n - 1$, satisfying equation (9) or the three respective encrypted values $B'_1(i)$, $B'_2(i)$ and $B'_3(i)$ satisfying equation (8).

Then, if M_1 and M_3 are given and we want to extract M_2, it is similar to say that $B_1(i)$ and $B_3(i)$ are given and we want to extract $B_2(i)$, we are interesting in solution of the above modular equation if $B_2(i)$ is not known. To extract $B_2(i)$, we have two cases:

First case: $B_1(i)$ and n are relatively primes. In this case, $B_1(i)$ has inverse modulo n and therefore the above equation possesses a single solution

modulo n. Thus, there is a single integer solution since $B_2(i)$ is supposed to be less than n, therefore:

$$B_2(i) \equiv B_1(i)^{-1} \times B_3(i) \; mod \; n. \tag{10}$$

Second case: $B_1(i)$ and n are not relatively primes. In this case, the only common divisors possible to $B_1(i)$ and n, are p and q. That is, $B_1(i)$ is multiple of p or q. Suppose that $B_1(i)$ is multiple of p, then $B_1(i) = k \times p$, for $k \in \{1, \ldots, q-1\}$. Thus, p divides $B_1(i)$ and n, and from the equation (9) necessarily p also divides $B_3(i)$. We can then write $B_3(i) = p \times \tilde{B}_3(i)$. The equation (9) signifies that there exist an integer l such:

$$k \times p \times B_2(i) = p \times \tilde{B}_3(i) + l \times p \times q, \tag{11}$$

and thus dividing by p gives:

$$k \times B_2(i) = \tilde{B}_3(i) + l \times q. \tag{12}$$

Thus, we have:

$$k \times B_2(i) \equiv \tilde{B}_3(i) \; mod \; q. \tag{13}$$

Since k is strictly less then q, it is relatively prime to q and thus invertible modulo q, therefore:

$$B_2(i) = k^{-1} \times \tilde{B}_3(i) \; mod \; q. \tag{14}$$

This single solution modulo q leads to p solutions for the block $B_2(i)$: one before q, one between q and $2q$ and so on; in the case of $B_1(i)$ is multiple of q, we have in the same way q solutions.

Since we would not have all single solutions for these noisy pixels of M_2, indeed a lot of blocks would be factor of the initial primes p and q, so they would give multiple solutions for each noisy block of M_2, and these solutions must be less than or equal to $\{1, \ldots, q\}$ and the original value of the noisy pixel of M_2 belongs to this solution set.

In order to select the best value from the solution set for the noisy pixel and to remove the noisy pixels from the extracted M_2, we take advantage of the homogeneity of the visual data, as usually there is high degree of coherence between the neighbors of image data. So we take mean of the non-noisy neighbors of noisy pixels of M_2 and this mean value is compared with each value of the solution set for the corresponding pixel, and then select the value from the solution set which is giving us the least distance from mean value.

4 Experimental Results and Discussions

We have tested the proposed algorithm on 200 gray level images (8 bits/pixel) of size 512×512 pixel. We have randomly partitioned the 200 gray level images into two groups (100 each), transferring image group and reconstruction image group, then we randomly selected two images M_1 and M_2, one for the transfer purpose

and second for reconstructed purpose. For our experimentation we have chosen the keys which follows the basic properties of RSA cryptosystem. We transformed each image into coefficient image where each coefficient is representing block of pixels using equation (6) and then encrypt each coefficient of the two images M_1 and M_2 with RSA by using equation (7).

After encryption of M_1 and M_2 we have scrambled the two corresponding encrypted images C_1 and C_2 by applying a multiplication modulo n to get a new scrambled image C_3. This scrambled image C_3 can be decrypted to produce M_3. These two images C_3 and M_3 are our intended images to be safe transferred by the sender to the receiver through insecure channel.

4.1 A full example

In Fig. 6 we visually present an example of the proposed method. Fig. 6.a and 6.b present two standard gray level original images of Lena and Barbara, each of size 512×512 pixels (8 bits/pixel), Fig. 6.c and 6.d illustrate the corresponding encrypted images and Fig. 6.e corresponds to the scrambled image from multiplication of the two encrypted images Fig. 6.c and 6.d. Finally, Fig. 6.f shows the resultant decrypted image of Fig. 6.e, which can be used for transfer purpose. Fig. 6.c-f are represented after decomposition of blocks in order to visualize pixel values.

In Fig.7, we show the extraction and reconstruction of the shared secret image. Fig. 7.a illustrates the extracted image with noisy pixels having multiple solutions corresponding to blocks of two pixels. Finally, Fig. 7.b shows the reconstructed image which is very near of the original image M_2. The peak signal to noise ratio (PSNR) between the original image, Fig. 6.a, and the reconstructed one, Fig. 7.b equals to 47.8 dB. This value shows high degree of resemblance between the original and the reconstructed image.

The strength and effectiveness of the proposed method applied to 100 images in terms of PSNR value between the original and the reconstructed images is shown in Fig. 8. and the mean value for the PSNR is 45.8 dB.

4.2 Comparison with XOR-based method

Exclusive-OR (XOR) is a binary operator which has the property that if it is applied between two numbers, and if one of the number is available after performing this operation on then we can get the second number by using resultant number and one of the two numbers. If we apply the XOR operation between two images M_1 and M_2 and transfer the resultant image through insecure channel, we can get any one of the image, if we have the second image: $M_{XOR} = M_1 \otimes M_2$ thus $M_1 = M_{XOR} \otimes M_2$ or $M_2 = M_{XOR} \otimes M_1$.

We can encounter two problems with this approach. First the resultant image M_{XOR} contains a lot of information about the two original images, for example if we applied the XOR operation between M_{XOR} and a homogeneous image (for example with all pixels equal to 128) then the resulted image can give a lot information about the two original intended images, as shown in Fig. 9.a while

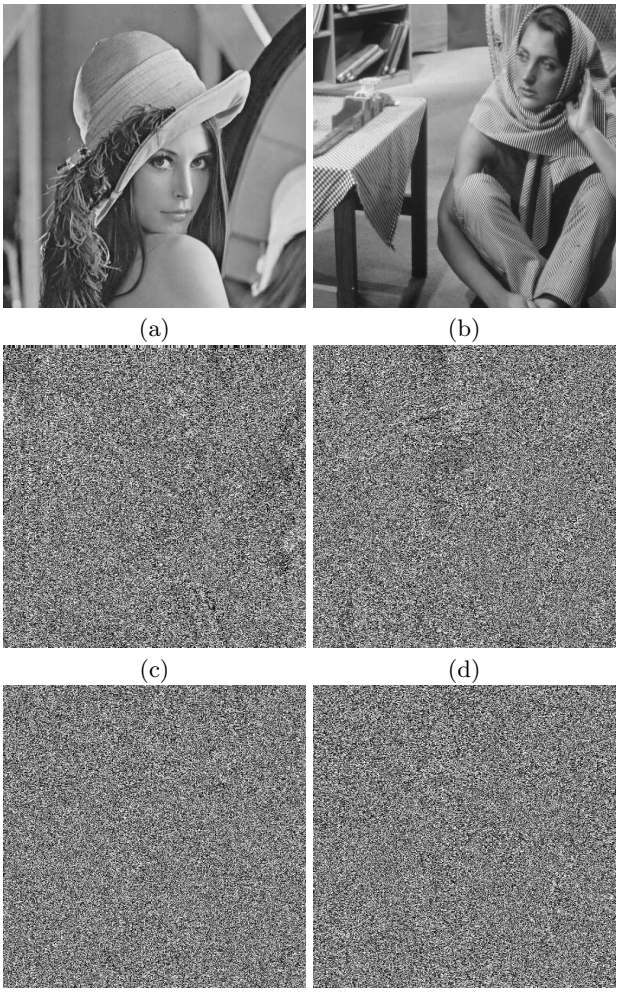


Fig. 6. a) and b) Original Images, c) Encrypted image of (a), d) Encrypted image of (b), e) Image obtained from multiplication of (c) and (d), f) Decrypted image of (e).

if the same homogeneous image is used as an attack on the proposed method we would have a resultant scrambled image with no worth-full information contents, as shown in Fig. 9.b.

The second problem is that XOR is not a homomorphic operator. Suppose we have encrypted images M_1 and M_2 and we apply XOR operation between C_1 and C_2 to produce C_{XOR}, now decrypting C_{XOR} gives M'_{XOR}, but when we apply XOR operation between M_{XOR} and M_1 the result does not produce the original image M_2.

(a)  (b)

Fig. 7. a) Extracted image, b) Reconstructed image.

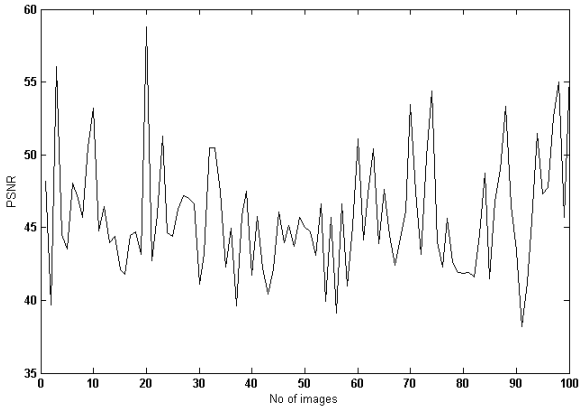


Fig. 8. Graphical display of PSNR values of 100 images.

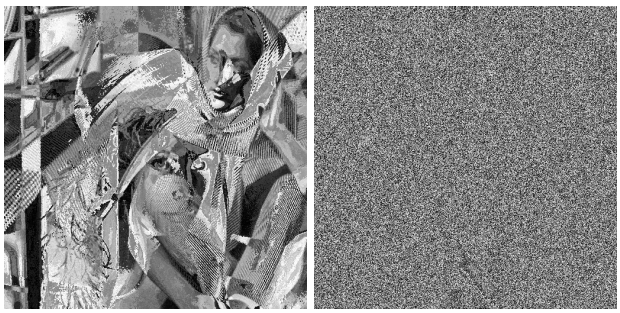
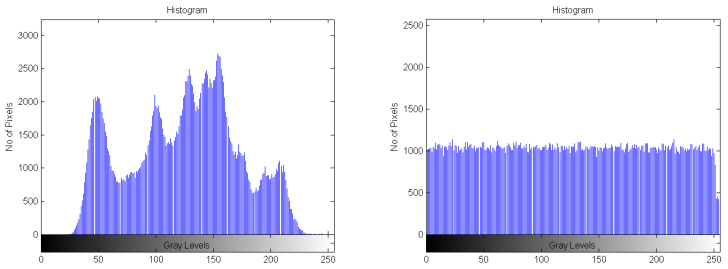
(a)  (b)

Fig. 9. a) Resultant image after attack by using a homogeneous image (grey level = 128) on XOR image M_{XOR}, b) Resultant image after attack by using a homogeneous image (grey level = 128) on the transferred image.

4.3 Security Analysis

Analysis of entropy and local standard deviation: The security of the encrypted images can be measured by considering the variations (local or global)



(a) (b)

Fig. 10. a) Histogram of original image of Lena, b) Histogram of scrambled image Fig. 6.e.

in the protected images. Considering this, the information content of image can be measured with the entropy $H(X)$, where entropy is a statistical measure of randomness or disorder of a system which is mostly used to characterize the texture in the input images. If an image has 2^k gray levels α_i with $0 \leq i \leq 2^k$ and the probability of gray level α_i is $P(\alpha_i)$, and without considering the correlation of gray levels, the entropy $H(X)$ is defined as:

$$H(X) = -\sum_{i=0}^{2^k} P(\alpha_i) log_2(P(\alpha_i)). \tag{15}$$

If the probability of each gray level in the image is $P(\alpha_i) = \frac{1}{2^k}$, then the encryption of such image is robust against statistical attacks, and thus $H(X) = log_2(2^k) = k$ bits/pixel. In the image the information redundancy r is defined as:

$$r = k - H(X). \tag{16}$$

When $r \approx 0$, the security level is acceptable. Theoretically an image is an order-M Markov source, with M the image size. In order to reduce the complexity, the image is cut in small block of size n and considered as an order-n Markov source. The alphabet of the order-n Markov source, called $X^{'}$ is β_i with $0 \leq i < 2^{k^n}$ and the order-n entropy $H(X^{'})$ is defined as:

$$H(X^{'}) = H(X^n) = -\sum_{i=0}^{2^{k^n}} P(\beta_i) log_2(P(\beta_i)). \tag{17}$$

We used $2^k = 256$ gray levels and blocks of n=2 or 3 pixels corresponding to a pixel and its preceding neighbors. In order to have minimum redundancy i.e. $r \approx 0$, in equation (16), we should have k=8 bits/pixel for equation (15) and k=16 or 24 bits/block for equation (17).

Similarly we also analyzed the variation of the local standard deviation $\sigma(j)$ for each pixel $p(j)$ taking account of its neighbors to calculate the local mean $\overline{p(j)}$, the formula for local standard deviation is given as:

$$\sigma(j) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(p(i) - \overline{p(j)})}, \tag{18}$$

where m is the size of the pixel block to calculate the local mean and standard deviation, and $0 \leq j < M$, if M is the image size.

In Fig. 10, we show the histogram of the original image of Lena and the histogram of the scrambled transmitted image, where the histogram of the transmitted safe image is different to the histogram of the original image. In Fig. 10.b, we can see a uniform distribution of the gray level values among the pixel coordinates of the transmitted image while in the histogram of original image Fig. 10.a, there is single blob of gray level values which signifies some shape or object. Similarly from equation (15) we get high entropy H(X) of 7.994 bits/pixel (H(X)= 7.45 bits/pixel for the original image of Lena). The information redundancy r, in equation (16) then equals to 0.006 bit/pixel. The order-2 entropy, $H(X^2)$ of equation (17) equals to 15.81 bits/block for Fig. 10.d (12.33 bits/block for the original image). The information redundancy r, is then less than 0.19 bit/block.

From equation (18) we also analyzed the variation of the local standard deviation σ for each pixel while taking its neighbors into account. The mean local standard deviation equals to 67.35 gray levels for the final scrambled image of Fig. 10.d, where as the mean local standard deviation equals to 6.21 gray levels for the original Lena image. These analysis show that the final scrambled image is protected against statistical attacks.

Correlation of adjacent pixels: Visual data is highly correlated i.e. pixels values are highly probable to repeat in horizontal, vertical and diagonal directions. Since RSA public-key cryptosystem is not random in nature, so it give same results for the same values of the inputs. It means that if an image region is highly correlated or having same values, then the public-key encryption will produce the same results, and a cryptanalyst can easily understand the information content related to the original image. A cryptosystem is considered robust against statistical attacks if it succeeds in providing low correlation between the neighboring pixels or adjacent pixels. The proposed encryption scheme generates a ciphered image with low correlation among the adjacent pixels. A horizontal correlation of a pixel with its neighboring pixel is given by a tuple (x_i, y_i) where y_i is the horizontal adjacent pixel of x_i. Since there is always three directions in images i.e. horizontal, vertical and diagonal, so we can define correlation in horizontal direction between any two adjacent pixels as:

$$corr_{(x,y)} = \frac{1}{n-1}\sum_{0}^{n}(\frac{x_i - \overline{x_i}}{\sigma_x})(\frac{y_i - \overline{y_i}}{\sigma_y}), \tag{19}$$

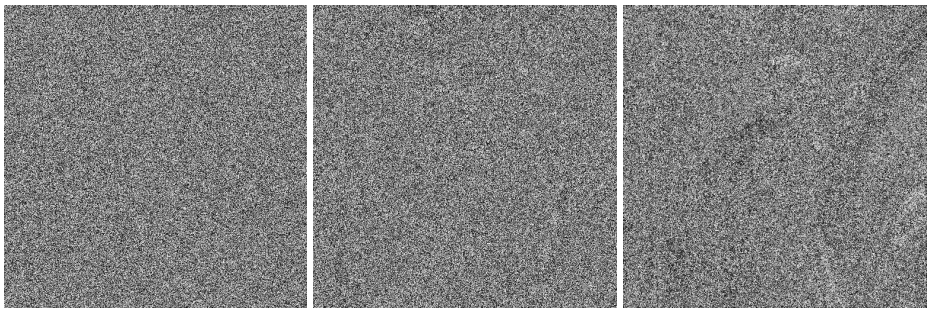
where n represents the total number of tuples (x_i, y_i), $\overline{x_i}$ and $\overline{y_i}$ represent the mean and σ_x and σ_y represent standard deviation respectively. In Table (1), we can see correlation values of Lena image and the transmitted scrambled image. It

can be noticed from the table that the proposed scheme retains small correlation coefficients in horizontal and vertical directions.

	Plain image	Encrypted image
Horizontal	0.9936	0.1693
Vertical	0.9731	-0.0010

Table 1. Correlation of horizontal and vertical adjacent pixels in two images

Key sensitivity test: Robustness against cryptanalyst can be improved if the cryptosystem is highly sensitive towards the key. The more the visual data is sensitive towards the key, the more we would have data randomness i.e. high value for the entropy and thus the lower we would have visual correlation among the pixels of the image. For this purpose, a key sensitivity test is assumed where we pick one key and then applied the proposed technique for encryption and then make a one bit change in the key and again applied the proposed encryption technique. Numerical results show that the proposed technique is highly sensitive towards the key change, that is, a totally different version of scrambled image is produced when the keys are changed, as shown in Fig. 11. Also from equation (19), we get a correlation value of 0.1670, which means there is negligible amount of correlation among the pixels of the ciphered image with different keys.



(a) (b) (c)

Fig. 11. Key sensitivity test: a) Encrypted image with key, K2, b) Image encrypted with K1 and decrypted with K2, c) Reconstructed image with key K2.

Also, if we encrypt an image with one key $K1$ and decrypt with a another key $K2$ and then apply the proposed scheme for the reconstruction of the original image, we can not get the original image, this observation can be seen in Fig. 11.b and 11.c.

5 Conclusions

In this paper, we proposed a method for sharing secret images during a transfer using carrier exploiting multiplicative homomorphic property of RSA algorithm. It has been observed that extraction of the original image from the transferred image is possible with the help of carrier image. For the reconstruction of the shared image, we have demonstrated that we have two particular cases. In the first case, we have a single solution and in the second case we have multiple solutions but only one corresponds to the original value. Experimental results showed that the reconstructed image after the extraction is visually indistinguishable of the original image. We can use this method on any public key cryptosystem satisfying multiplicative or additive homomorphic property.

References

1. J.C. Borie, W. Puech, and M. Dumas. Encrypted Medical Images for Secure Transfer. In *ICDIA 2002, Diagnostic Imaging and Analysis, Shanghai, R.P. China*, pages 250–255, Aug. 2002.
2. C. Fontaine and F. Galand. A Survey of Homomorphic Encryption for Nonspecialists. *EURASIP Journal Information Security*, 2007(1):1–15, 2007.
3. El Gamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete logarithms. *IEEE Transactions on Information Theory*, pages 469–472, 1985.
4. M. Kuribayashi and H. Tanaka. Fingerprinting Protocol for Images Based on Additive Homomorphic Property. *IEEE Transactions on Image Processing*, 14(12):2129–2139, Dec. 2005.
5. P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residousity Classes. *(Springer-Verlag)*, 1592:223–238, 1999.
6. S. R. M. Prasanna, Y. V. Subba Rao, and A. Mitra. An Image Encryption Method with Magnitude and Phase Manipulation using Carrier Images. *International Journal of Computer Vision*, 1(2):132–137, 2006.
7. W. Puech and J.M. Rodrigues. A New Crypto-Watermarking Method for Medical Images Safe Transfer. In *Proc.* 12^{th} *European Signal Processing Conference (EUSIPCO'04)*, pages 1481–1484, Vienna, Austria, 2004.
8. D.K. Rappe. Homomorphic Cryptosystems and their Applications. Cryptology ePrint Archive, Report 2006/001, 2006.
9. B. Schneier. *Applied cryptography.* Wiley, New-York, USA, 1995.
10. A. Uhl and A. Pommer. *Image and Video Encryption: From Digital Rights Management to Secured Personal Communication.* Springer, 2005.