Application of Homomorphism to Secure Image Sharing

Naveed Islam, William Puech, Khizar Hayat, Robert Brouzet

Optics Communications 284 (2011) 4412–4429

Contents lists available at ScienceDirect

Optics Communications

journal homepage: www.elsevier.com/locate/optcom

Application of homomorphism to secure image sharing

Naveed Islam [a], William Puech [a,*], Khizar Hayat [b], Robert Brouzet [c]

[a] LIRMM Laboratory, UMR CNRS, 5506, University of Montpellier II, 161 rue Ada, 34392 Montpellier, France
[b] Department of Computer Science, COMSATS Institute of Information Technology, University Road, 22060 Abbottabad, Pakistan
[c] I3M Laboratory, UMR CNRS 5149, University of Montpellier II, 34392 Montpellier, France

ARTICLE INFO

Article history:
Received 5 November 2010
Received in revised form 8 April 2011
Accepted 30 May 2011
Available online 22 June 2011

Keywords:
Cryptosystem
Homomorphism
Image encryption
RSA
Paillier
Security

ABSTRACT

In this paper, we present a new approach for sharing images between l players by exploiting the additive and multiplicative homomorphic properties of two well-known public key cryptosystems, *i.e.* RSA and Paillier. Contrary to the traditional schemes, the proposed approach employs secret sharing in a way that limits the influence of the dealer over the protocol and allows each player to participate with the help of his key-image. With the proposed approach, during the encryption step, each player encrypts his own key-image using the dealer's public key. The dealer encrypts the secret-to-be-shared image with the same public key and then, the l encrypted key-images plus the encrypted to-be shared image are multiplied homomorphically to get another encrypted image. After this step, the dealer can safely get a scrambled image which corresponds to the addition or multiplication of the $l+1$ original images (l key-images plus the secret image) because of the additive homomorphic property of the Paillier algorithm or multiplicative homomorphic property of the RSA algorithm. When the l players want to extract the secret image, they do not need to use keys and the dealer has no role. Indeed, with our approach, to extract the secret image, the l players need only to subtract their own key-image with no specific order from the scrambled image. Thus, the proposed approach provides an opportunity to use operators like multiplication on encrypted images for the development of a secure privacy preserving protocol in the image domain. We show that it is still possible to extract a visible version of the secret image with only l-1 key-images (when one key-image is missing) or when the l key-images used for the extraction are different from the l original key-images due to a lossy compression for example. Experimental results and security analysis verify and prove that the proposed approach is secure from cryptographic viewpoint.

1. Introduction

With each passing day, digital communication is becoming increasingly vulnerable to malicious interventions or monitoring like hacking or eavesdropping. The security of sensitive visual data in applications, like safe storage, authentication, copyright protection, remote military image communication or confidential video-conferencing, requires new strategies for safe transmission over insecure channel. There are two common techniques used for the secure transmission of data, namely cryptography and steganography. The confidentiality can be ensured making the message unreadable, with the help of secret keys, with cryptography [1] and with steganography by hiding the message in some innocent carrier signal [2]. Homomorphic cryptosystems are a special type of cryptosystems that preserves 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 [3]. Homomorphic property of public key cryptosystems has been employed in various data security protocols like electronic voting system, bidding protocols, cashing systems and asymmetric fingerprinting of images [4].

The classical approach of applying cryptographic techniques to visual data, *e.g.* an image, is to convert each pixel or block of pixels of the image into a ciphered value and transmit it through some channel, not necessarily secure, towards the receiver. The receiver applies the decryption algorithm on the encrypted image to transform it back to its original pixel form. Due to the large size of the visual data, larger storage capacities, computation time and higher transmission bandwidths are required. Keeping these three factors in view, most of the existing systems rely on symmetric cryptosystem which, due to the use of limited key size, produces a ciphered image that not only requires low computation time during encryption or decryption but also lower memories relative to the asymmetric cryptosystems [5–7]. Asymmetric cryptosystems usually employ relatively larger key sizes for data encryption, resulting in higher memory and computational costs. That is why even today they are employed for tasks like key management and authentication.

* Corresponding author.
E-mail addresses: naveed.islam@lirmm.fr (N. Islam), william.puech@lirmm.fr (W. Puech), khizarhayat@ciit.net.pk (K. Hayat), robert.brouzet@unimes.fr (R. Brouzet).

0030-4018/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.optcom.2011.05.079

For the shared trust and distributed environment, secret sharing schemes provide sufficient security in various communication applications. Secret sharing is a method to divide a secret S into certain l number of shares *i.e.* $s_1, \ldots, s_l$, where each individual share s_i does not have any information about the secret S. The secret S can be reproduced if k shares (with $k \leq l$) are combined. Traditional secret sharing schemes can be traced back to the (k, l) threshold scheme which partitions the secret message into l shares and requires at least k or more shares for the reconstruction. Secret sharing over visual data is called Visual Secret Sharing (VSS), where the secret image is required to be shared among a group of players and all the shares are needed in the construction of the secret image [8,9]. The problem with traditional secret sharing scheme is the possibility of dishonest dealer, who may favor some individual or a group of players or may provide a wrong share to some players. Similarly, the players are only at the receiving and decryption end and lacks of player participation in the encryption step limit the capabilities of the secret sharing scheme in many applications.

Unlike the traditional secret sharing schemes, the proposed scheme does not make shares of the original secret image but constructs a combined secret from various key-images from the players and there is only one resultant image which is to be shared among all the players of the secret. Thus, this approach limits the dealer's contribution in the creation of the secret shares and allows the player participation for secret sharing trust. In this paper, we provide two different applications, one employing the additive homomorphic property of the Paillier cryptosystem [10] and the other relying on the multiplicative homomorphic property of the RSA cryptosystem [1] to share a secret image using l key-images for the sharing, transfer and extraction of the original secret image. With the proposed approach, the processing time to extract the secret shared image is very low because it is based on simple arithmetic operations. Moreover, the scrambled shared image preserves the original size. We will also notice that the dealer does not intervene in the extraction process and no specific order is followed while using the key-images of the players.

This paper is organized as follows. In Section 2, we give an introduction of Paillier and RSA with special reference to their homomorphic property and we explain how to apply it to an image. The proposed algorithms are detailed in Section 3, which begins with an application scenario, followed by general steps of the proposed approach. Due to the multiplicative homomorphic property of RSA, the extracted image could have noisy pixels, therefore the probability distribution of these noisy pixels and their effects is also presented in Section 3. Experimental results along with the security analysis of the proposed scheme are studied in Section 4. Finally, Section 5 gives a brief summary and concluding remarks.

2. Previous work

We denote by Z_n the set of integers modulo n, by Z_n^* the set of invertible elements modulo n, *i.e.* all integers that are relatively prime to n, by $gcd()$ the greatest common divisor, by $lcm()$ the least common multiple, by $E()$ the encryption function, by $D()$ the decryption function, by $\phi(n)$ the total number of integers less than and relatively primes to n, by M the message that is partitioned into the blocks $m(i)$, and by C the ciphertext that is partitioned into blocks $c(i)$ in correspondence to $m(i)$. In this paper, the term *encryption* refers to the use of cryptographic encryption while the term *scrambling* refers to other techniques which make the signal lose its significance.

This section introduces some classical cryptographic protocols, including both the symmetric and asymmetric cryptosystems. Paillier and RSA cryptosystems are explained in detail with special reference to their homomorphic character. Thereafter, we discuss the use of cryptography in image domain along with the partitioning of image into blocks for the use of larger key sizes for security. Finally, we present the standard protocol for image transmission and then give a brief description of VSS schemes.

Extra storage capacities and special computation are required for multimedia data, like images, videos or 3D objects, due to the involvement of huge amount of data. Various cryptographic techniques are used for the secure transfer of multimedia data. Modern image based cryptographic techniques may involve full encryption or selective encryption of the image, depending on the application. Since many applications require real time performance, partial encryption is mostly preferred [11]. Encryption approaches can be divided into symmetric-key cryptosystems or asymmetric-key cryptosystems. In symmetric cryptosystems, the same key is used for both encryption and decryption. Symmetric key cryptosystems are usually very fast and easy to use. Since the same key is used for encryption and decryption, the key needs to be securely shared between the emitter and the receiver. In asymmetric cryptosystem, two different keys are required: the public and the private keys. With the receiver's public key, the sender encrypts the message and sends it to the receiver who decrypts the message with his private key. Some known algorithms are RSA [1], Paillier cryptosystems [10] and El Gamal [12]. RSA and El Gamal are public-key cryptosystems which support the homomorphic operation of multiplication modulo n whereas Paillier cryptosystem supports the homomorphic addition of encrypted messages.

2.1. Paillier encryption scheme

Pascal Paillier proposed a cryptosystem which is based on composite degree residousity class problem [10]. The public and private keys are generated as follows. Let $n = p \times q$, where p and q are two large and different prime numbers such that $gcd(n, \phi(n)) = 1$, calculate $\lambda(n) = lcm(p-1, q-1)$ and choose $g \in Z_{n^2}^*$ such that $gcd(L(g^{\lambda(n)} mod\ n^2), n) = 1$, where $L(t) = \frac{(t-1)}{n}$. The public key is composed of (n, g) and the private key is composed of $\lambda(n)$. Thus, the message space is represented by Z_n and the cipher space is represented by $Z_{n^2}^*$, which means that the size of the cipher space is square of the size of the message space.

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 $c(i)$. Thus, given a message block $m(i)$ with $0 \leq m(i) < n$, and a public key (n, g), choose a random number $r_i \in Z_n^*$, then the encryption $c(i)$ of $m(i)$ is given by:

$$c(i) = E(m(i)) \equiv g^{m(i)} r_i^n \ mod\ n^2. \tag{1}$$

Given a ciphertext $c(i)$ with $0 \leq c(i) < n^2$ and a private key $\lambda(n)$, the decryption $m(i)$ of $c(i)$ is given by:

$$m(i) = D(c(i)) \equiv \frac{L\left(c^{\lambda(n)} \ mod\ n^2\right)}{L\left(g^{\lambda(n)} \ mod\ n^2\right)} mod\ n. \tag{2}$$

Cryptosystems are either deterministic or probabilistic: deterministic cryptosystem produces the same ciphertext every time for the same plaintext and keys while probabilistic cryptosystem, like Paillier, includes a random number r_i which produces different values for the ciphertext providing the same plaintext.

Example: assume primes p and q are given as $p = 7$, $q = 11$, then $n = p \times q = 77$. Let $g = 2$, $r_1 = 5$, $r_2 = 6$ and let the two messages be $m_1 = 4$ and $m_2 = 5$. Then, the encryption of m_1 is given as: $c_1 \equiv g^{m_1} \times r_1^n \ mod\ n^2 = 2^4 \times 5^{77} mod\ 77^2 = 3436$ and the encryption of m_2 is given as: $c_2 \equiv g^{m_2} \times r_2^n \ mod\ n^2 = 2^5 \times 6^{77} \ mod\ 77^2 = 4623$.

2.2. RSA cryptosystem

RSA is a well known asymmetric cryptosystem, developed in 1978 [1]. The main procedure consists of selecting two large and different

prime numbers p and q, calculating their product $n=p\times q$ and selecting an integer e, which is relatively prime to $\phi(n)$ and with $1<e<\phi(n)$, where $\phi()$ 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 is (d). For the encryption, the plaintext M is also partitioned into blocks $m(i)$ such that $m(i)<n$ and for each plaintext $m(i)$ we get a ciphertext $c(i)$:

$$c(i) = E(m(i)) \equiv m(i)^e \ \ mod\ \ n. \tag{3}$$

For the decryption, with the ciphertext $c(i)$ we can obtain the original plaintext $m(i)$ by the equation:

$$m(i) = D(c(i)) \equiv c(i)^d \ \ mod\ \ n. \tag{4}$$

Example: assume primes p and q are given as $p=7$, $q=17$, then $n=p\times q=119$. If $e=5$, then $gcd(\phi(119), 5)=1$ and we get $d\equiv e^{-1} mod\ \phi(n)=77$. Let the input message be $m_1=22$ and $m_2=19$, then the encryption of m_1 is given as: $c_1\equiv 22^5 mod\ 119=99$ and the encryption of m_2 is given as: $c_2\equiv 19^5 mod 119=66$.

2.3. Homomorphism

Most of the well known asymmetric cryptosystems follow either additive **homomorphism** or multiplicative **homomorphism**. A function is said to be homomorphic if it obeys the following condition:

$$f(x\oplus y) = f(x)\otimes f(y), \tag{5}$$

where $\otimes$ and $\oplus$ may be any distinct arithmetic operations. In Cryptography, any encryption algorithm $E()$ is said to be homomorphic if, given $E(m_x)$ and $E(m_y)$, one can obtain $E(m_x\oplus m_y)$ without decrypting $E(m_x)$ and $E(m_y)$ [13]. For $E()$, with Eq. (5) we have then:

$$E\left(m_x\oplus m_y\right) = E(m_x)\otimes E\left(m_y\right), \tag{6}$$

where $\oplus$ and $\otimes$ can be addition or multiplication. Usually the $\oplus$ operation is either addition or multiplication while the $\otimes$ operation is multiplication. A decryption function $D()$ is said to be homomorphic if:

$$D\left(E(m_x)\otimes E\left(m_y\right)\right) = D\left(E\left(m_x\oplus m_y\right)\right), \tag{7}$$

and then:

$$D\left(E(m_x)\otimes E\left(m_y\right)\right) = m_x\oplus m_y. \tag{8}$$

With the algorithm of Paillier, the decryption obeys the following homomorphic property: $\forall m(i)\in Z_n$ and $i\in N$,

$$D\left(\prod_{i=1}^{p} E(m(i)) mod\ \ n^2\right) \equiv \sum_{i=1}^{p} m(i) mod\ \ n. \tag{9}$$

Due to additive homomorphic property, Paillier cryptosystem is widely used in electronic voting system.

Similarly, the decryption function of the RSA algorithm obeys the following multiplicative homomorphism:

$$D\left(\prod_{i=1}^{p} E(m(i)) mod\ \ n\right) \equiv \prod_{i=1}^{p} m(i) mod\ \ n. \tag{10}$$

For the proposed approach, presented Section 3, we have used these two properties.

Example (Paillier): with the values of the example presented in Section 2.1 we have $c_3=c_1\times c_2\equiv 3436\times 4623\ mod\ 77^2=837$. Applying the decryption algorithm over c_3 results in 9, which is equivalent to the addition of the two plaintexts *i.e.* $m_3=m_1+m_2\equiv 4+5\ mod\ 77=9$. Hence Paillier supports homomorphic operation of addition modulo n^2 over the plaintext, presented in Eq. (9).

Example (RSA): with the values of the example presented in Section 2.2 we have $c_3=c_1\times c_2\equiv 99\times 66\ mod\ 119=108$. Applying the decryption algorithm over c_3 results in 61, which is equivalent to the multiplication of the two plaintexts *i.e.* $m_3=m_1\times m_2\equiv 22\times 19\ mod\ 119=61$. Hence RSA supports homomorphic operation of multiplication modulo n, presented in Eq. (10).

2.4. Image encryption

If one considers the message space of an image limited to the size of the pixel and does not want to increase the size of the encrypted image, then the required key size is very limited. This makes the secured image vulnerable to any brute force attack and an adversary may be able to retrieve the key. Contrary to the limited-sized key, if higher level security is provided through standard key size like 1024 bits, the overall size of the encrypted image may increase up to the key-sized multiple its initial size, depending on the cryptosystem involved. Extreme care must be taken while calculating the values of the keys because the quality of the encrypted image depends on the value of the public key and bad keys can produce encrypted images which resemble the original image [14]. In the block-based image encryption scheme, as described in Section 2 for plaintext M, the image is partitioned into equal-sized blocks, depending on the size of the key. The creation of block and encryption should be carried out in such a way that the encrypted image does not reveal any structural information about its original contents. Other recent approaches in image security are based on chaotic functions, which are characterized by ergodicity (stochastic behavior of image), extreme dependence on initial condition and randomness in the image [15–17]. The use of carrier image for the encryption of image has been presented in [18] using private key cryptosystem in the frequency domain.

If the length of the encryption key is γ bits, then the number of pixels, b, in each block is given by:

$$b = \lceil \gamma / k \rceil, \tag{11}$$

where k is the number of bits in a single pixel.[1] Let an image composed of $H\times L$ pixels $p(i)$, where $0\le i<H\times L$, the construction of the block values from the original image pixels $p(i)$ is given as:

$$m(i) = \sum_{j=0}^{b-1} p(i*b+j)\times 2^{kj}, \tag{12}$$

where $0\le i<\lceil H\times L/b\rceil$ for the block-partitioned image.

For Paillier cryptosystem, to be applied on each block, let $m(i)$ be the ith constructed block of an image, then the encryption of $m(i)$ is given by Eq. (1), where $m(i)$ is coded on γ bits and $c(i)$ is coded on 2γ bits. Similarly for RSA cryptosystem, the encryption of $m(i)$ is given by the Eq. (3), where $m(i)$ and $c(i)$ both are coded on γ bits.

After the decryption, the extraction of the original pixels from the transformed blocked image is given by:

$$\begin{cases} if & j=0 \\ & p(i\times b+j) \equiv \ m(i) mod\ 2^k \\ else & \\ & p(i\times b+j) \equiv \ m(i) mod\ 2^{k(j+1)} \\ & \qquad\qquad -\sum_{l=0}^{j-1} p(l))/2^{kj} \end{cases}. \tag{13}$$

[1] For example: 8 bits for a gray level image.

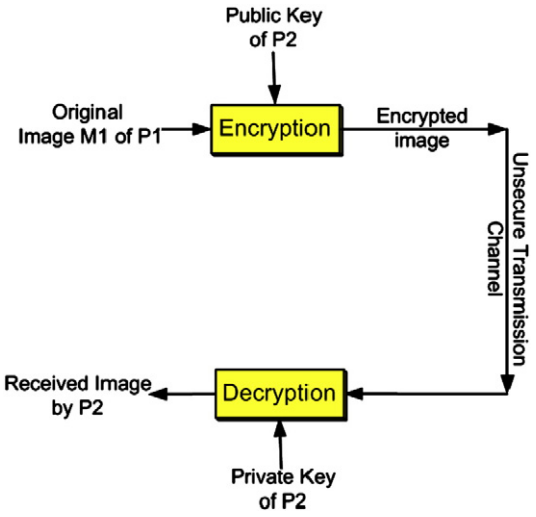

Fig. 1. Standard way for image transmission.

2.5. Standard protocol for image transmission

The standard protocol for secure data transfer is based on the security of the keys. In a standard asymmetric procedure, if a user P_1 wants to send an image M_1 to user P_2, he will first encrypt the image with the public key of the receiver *i.e.* P_2. This encrypted image will be then transmitted to the user P_2 over a transmission channel, not necessarily secured. At the receiving end, in order to read the image, the user P_2 will decrypt the image with his private key, as shown in Fig. 1.

In the case where l players want to share and transmit an encrypted image, a naive way would require l keys for the encryption of the secret image. Each player will encrypt the image with his private key and for the decryption of the secret shared image, each player will be required to use his private key to decrypt the secret image. The major disadvantage is the computational cost involved at both the sending and receiving ends, as shown in Fig. 2.

2.6. Visual cryptography

Visual cryptography, introduced by Naor and Shamir [9], involves the encoding of a secret image S into l shared images. These shared images are distributed among the players but each of them does not reveal any information about the secret image. Moreover, the secret image can only be revealed if the l shared images are stacked. In (k, l)-threshold scheme, the secret S can only be reconstructed from any k out of l shares but not fewer. The (k, l)-threshold scheme was further extended to general access structures when the concept of qualified sets and forbidden sets were introduced [19]. Here a set $T=(1, \ldots, l)$ is called the participant set, which is partitioned into two subsets called the qualified $t(1)_{qual}$ and the forbidden $t(2)_{forb}$ sets, such that, $t(1)_{qual} \cap t(2)_{forb} = \phi$ and the pair$(t(1), t(2))$ is called the general access structure of the scheme. The minimal qualified set is defined as: $Min(qualified) = \{A \in t(1)_{qual} : A' \notin t(1)_{qual}, \forall A' \subset A\}$.

VSS schemes were mostly applied on a single grayscale image but further developments on multiple secret image sharing have been proposed in [20]. One of the characteristics of the traditional visual secret sharing schemes was that the reconstructed image had a lot of noise, *i.e.* the reconstructed image was not 100% the original secret image and that is why a number of methods were introduced for the quality optimizations of the decrypted images [21]. Similarly the use of multiple grayscale or color secret image in VSS schemes was studied in [22–24].

Unlike VSS, the proposed approach is based on the contribution of secret shares of all the players for the creation of a secret sharing protocol using some public key cryptosystem. During the extraction of the secret image, all the l secret shares are required which is analogous to (l, l) scheme in VSS. It must be noted that for additive homomorphic scheme the proposed approach can be used as $(l-1, l)$ scheme for the extraction of a missing key-image being utilized in the secret sharing process. Also the (l, l) scheme can be modified to (l, l) scheme in additive homomorphism, where l represents any visual distortions (such compression, filtering, noise, …) in the key-images during the process of extraction (presented Section 4.1). In the proposed approach, presented in Section 3, the dealer wants to share a secret image among l players, but to extract the shared image, the l players do not need the dealer.

3. The proposed encryption method

In this section we first give a scenario where the proposed approach can be used. This is followed by an overview of the proposed approach which includes the encryption and decryption steps followed by the extraction of the secret shared image. We discuss the problems of multiplicative homomorphic cryptosystem of RSA during the extraction process, which can lead to visual loss of pixels (noisy pixels) in the extracted shared secret image. We analyze the probability distribution of such noisy pixels, due to RSA cryptosystem, for various block sizes.

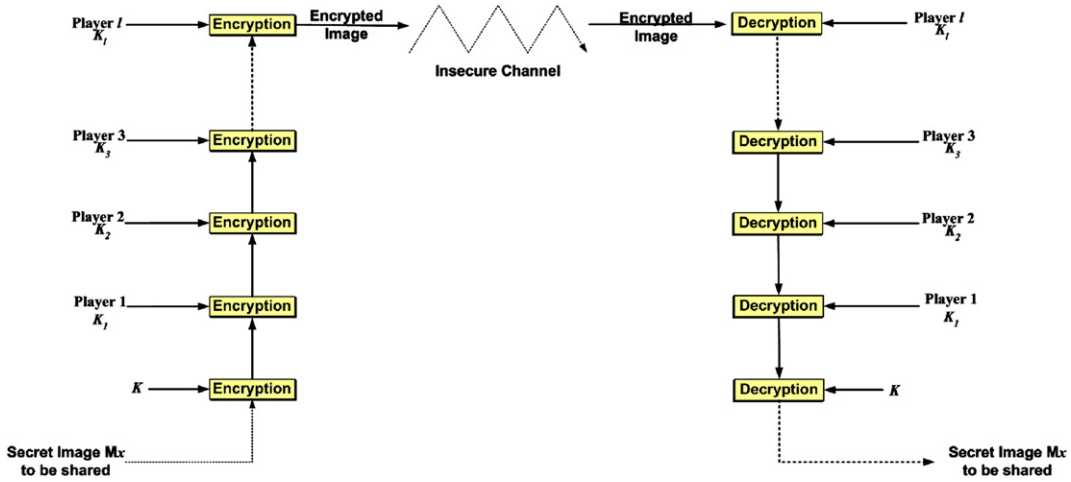

Fig. 2. A naive approach for encryption and decryption of shared secret image.

3.1. Scenario

Assume that a secret image is to be shared by a dealer among a group of l remote skeptic players. Since the players and the dealer do not trust each other, traditional secret sharing schemes which require the dealer to make share of the secrets, would not be applicable in this scenario. We assume that each remote player has his own secret image called **key-image**, which is used in the creation of the proposed shared secret. The proposed secret sharing scheme makes a single share which is given to each player as a secret shared image. The secret image can be reconstructed when the shared secret image plus the l key-images are combined together. If modern cryptographic approaches are applied, the naive way would require l keys for the encryption and the decryption of the shared secret image as described in Section 2.5.

To accomplish this secret sharing, the dealer transmits to each player a scrambled shared secret image which is constructed from all the key-images of the l players. The l players can then together construct the secret image. Note that the processing time is decreased during the extraction of the secret image because no decryption algorithm is required. We can also notice that the dealer has no role during the extraction and the secret shared image has preserved the original size.

3.2. Overview

As described in the scenario, the purpose of the proposed scheme is to securely share a secret image among a group of l players and even if an intruder gets a copy of the protected shared image, he must not be able to extract the original secret image. Also, at the receiving end, traditional arithmetic operation is used for the extraction of the secret image. This reduces the load of decryption at each step. Thus computation gain at the receiving end is achieved due to the use of key-images. Let a secret image be shared among l participants, where each of the l participants has his own key-image and all the images have the same size. We take these $l+1$ images –M_1, …, M_l and the secret image to be shared, M_x– and partition each image into blocks by using Eq. (12). With the public key given by the dealer, an asymmetric encryption is applied on each block of the $l+1$ transformed images. Note that the same public key is used for the individual encryption of the $l+1$ images.

After all the $l+1$ images have been encrypted, the proposed approach takes modulo multiplication of the $l+1$ encrypted images in a specific order so that none of the individual secret image is exposed to any other participant and no information about the number of the participants are revealed and thus get another encrypted image C_y. Because of the homomorphic property, this encrypted image must be the same if we had first applied any arithmetic operation like addition or multiplication over the $l+1$ original images to get a scrambled image and then apply a homomorphic encryption algorithm. This encrypted image C_y is decrypted to get a scrambled image M_y which is transferred over any insecure channel, to be shared among the recipients. Since M_y contains components of all the l key-images and the secret original image M_x, one can extract any one of the $l+1$ original images if l key-images are available. At the receiving end, as we have the l key-images and we have the scrambled image M_y, we can then extract the secret original image M_x.

3.3. Encryption step

For each block of the $l+1$ images M_1, …, M_l and M_x, the proposed scheme can be developed with the two homomorphic encryption schemes presented in Section 2, Paillier and RSA, using Eqs. (1) and (3).

The individual encryption of the $l+1$ original images *i.e.* M_1, …, M_l (l key-images) and M_x (secret image) will result into $l+1$ encrypted images C_1, …, C_l and C_x, respectively. Incremental multiplication of each encrypted image with another encrypted image results in a new encrypted image, which is fed-forward into the homomorphic multiplication process till the secret encrypted image M_x is input to give the final encrypted image C_y in the end, as shown in Fig. 3. Thus all the individual encrypted images are used in the process of securing the secret image M_x but no individual key-image is revealed. Since two different encryption schemes can be used, the size of each block

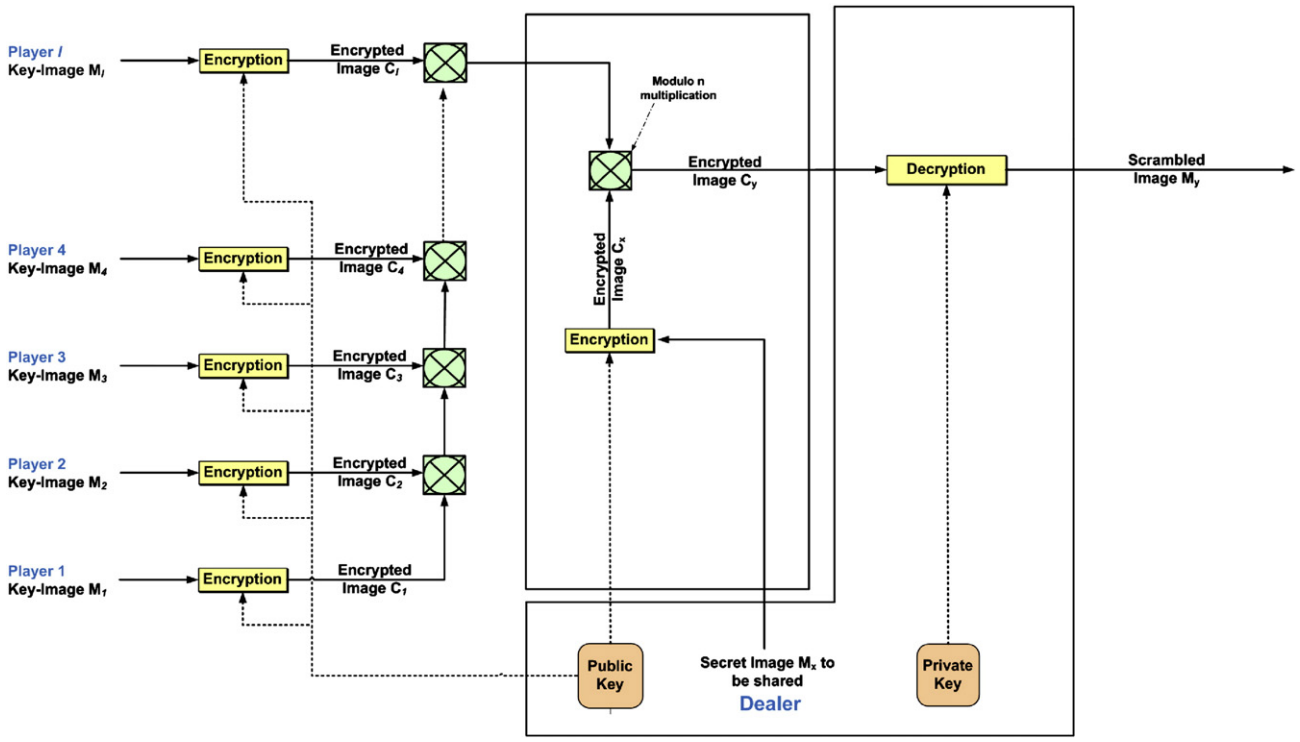

Fig. 3. Overview of proposed method.

of the resultant encrypted image could also be different. Thus, if the proposed scheme is applied over Paillier cryptosystem, each block of the $l+1$ encrypted images $C_1, \ldots, C_l$ and C_x would have values between 0 and $2^{2\gamma}-1$ while for RSA cryptosystem the block values would be between 0 and $2^{\gamma}-1$. After the step forward multiplication of all the $l+1$ encrypted images, modulo operation must be applied to get scrambled blocks of C_y, codable on γ and 2γ bits. For Paillier algorithm, the following equation is followed for the modulo operations:

$$c_y(i) \equiv \left(\prod_{l=1}^{l} c_l(i) \times c_x(i) \right) mod\ n^2. \tag{14}$$

For RSA the following equation is followed for the modulo operations:

$$c_y(i) \equiv \left(\prod_{l=1}^{l} c_l(i) \times c_x(i) \right) mod\ n. \tag{15}$$

Note that the modulo operation is based on n^2 while using Paillier cryptosystem. The encrypted image C_y is built from the blocks $c_y(i)$ and its decrypted image M_y is our intended image to be transferred or shared through some channel, not necessarily secured.

3.4. Decryption and extraction for Paillier scheme

The block diagram for the decryption and extraction by the proposed method, using the Paillier cryptosystem, is shown in Fig. 4. At the receiving end, we have $M_1, \ldots, M_l$ and M_y and we want to extract M_x. Due to the additive homomorphic property of Paillier, we have from Eqs. (9) and (14):

$$m_y(i) \equiv \left(\sum_{l=1}^{l} m_l(i) + m_x(i) \right) mod\ n. \tag{16}$$

We can compute inverse modulo of Eq. (16), which gives unique values for the blocks $m_x(i)$ of the image M_x. Now if $M_1, \ldots, M_l$ and M_y are given and we want to extract M_x, it is similar to say that $m_1(i), \ldots, m_l(i)$ and $m_y(i)$ are given and we want to extract $m_x(i)$, *i.e.* we are interested in solution of a modular equation if $m_x(i)$ is not known. Since $m_1(i), \ldots, m_l(i)$ and $m_y(i)$ are given, then from Eq. (16) we have:

$$m_x(i) \equiv \left(m_y(i) - \sum_{l=1}^{l} m_l(i) \right) mod\ n. \tag{17}$$

With Paillier cryptosystem, the approach is totally reversible, we can extract the secret shared image without loss. As illustrated in Fig. 4, the processing time is decreased during the extraction of the secret image because no decryption algorithm is carried out. We can also notice that the dealer does not intervene in the extraction process and no specific order of using the key-images of the players is necessary.

3.5. Decryption, extraction and reconstruction for RSA scheme

3.5.1. Extraction of the best solution

The process of decryption and extraction using RSA is analogous to Paillier cryptosystem in Fig. 4 but since RSA is multiplicative homomorphic, the operations in Fig. 4 are changed to multiplication. At the receiving end, we have $M_1, \ldots, M_l$ and M_y and we want to extract M_x. Due to the multiplicative homomorphic property of RSA, we have from Eqs. (10) and (15):

$$m_y(i) \equiv \left(\prod_{l=1}^{l} m_l(i) \times m_x(i) \right) mod\ n. \tag{18}$$

The extraction of secret image $m_x(i)$ is given as:

$$m_x(i) \equiv \left(m_y(i) \times \prod_{l=1}^{l} m_l(i)^{-1} \right) mod\ n. \tag{19}$$

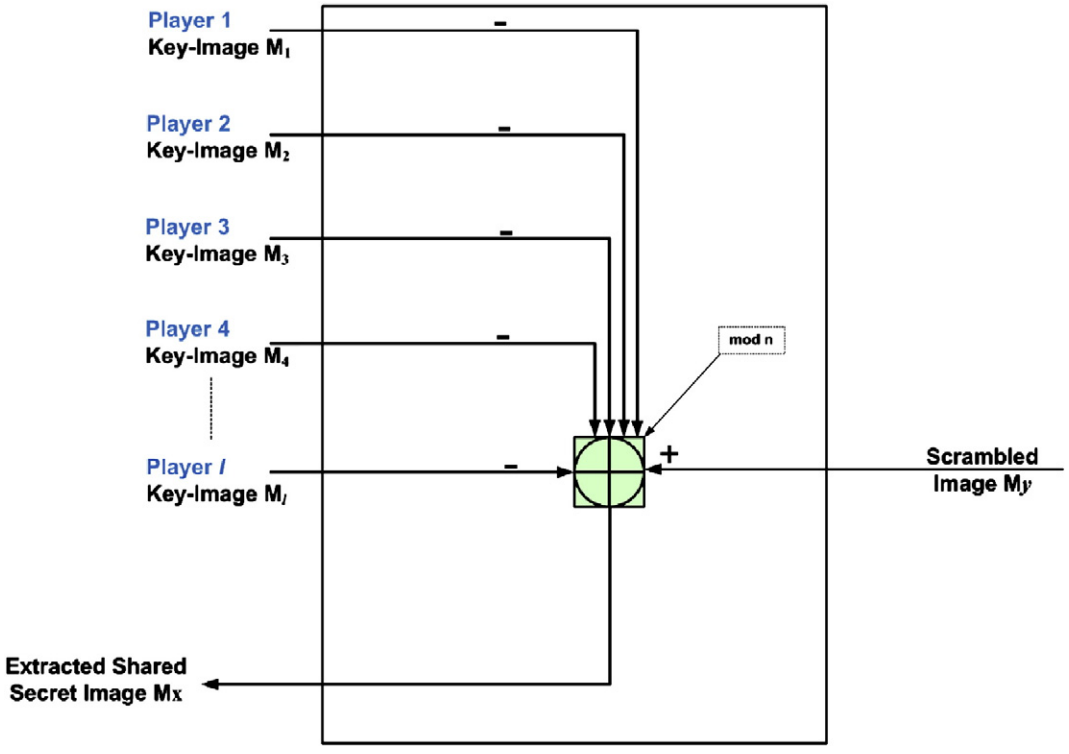

Fig. 4. Decryption and extraction of the protected shared image.

The result presented in Eq. (19) holds only when the initial data set do not contain factors of the prime product $n = p \times q$. Indeed, during the multiplicative inverse operation multiple solutions for the extraction can appear. It means that multiplicative homomorphic cryptosystem generates noise in the decryption and extraction process due to prime multiplicity.

In order to explain the prime multiplicity problem in multiplicative homomorphic cryptosystem, we try to apply the proposed scheme on two images which have pixel-block values which are factors of the prime product. We assumed that a secret image M_x is to be shared or transferred through insecure channel and a key-image M_1 is used as a carrier image for security. After the application of the proposed scheme, a third image M_y is received which is now the secret shared image. Now if the key-image M_1 and shared image M_y are both available, the extraction of the secret image M_x is possible, but this extracted image M_x will have missing or noisy pixels which will require post processing. The block diagram of the proposed method using the RSA cryptosystem for extraction and reconstruction for two images is shown in Fig. 5. Due to the multiplicative homomorphic property of RSA, according to Eq. (18) we have:

$$m_y(i) \equiv m_1(i) \times m_x(i) mod\ \ n. \tag{20}$$

We can do inverse modulo operation on Eq. (20), which gives single values for the blocks $m_1(i)$ of M_1 which are relatively prime to n and multiple solutions for the blocks $m_1(i)$ which are not relative primes to n. The reconstruction step consists in choosing the best value among the multiple solutions for particular pixels in order to reconstruct the original image M_x.

In order to explain the principles that make the extraction of the second image M_x possible, let us consider that p and q are primes such that $p<q$, and $n = p \times q$. Let $m_1(i)$, $m_x(i)$ and $m_y(i)$ three integers between 0 and n-1, satisfying Eq. (20). Then, if M_1 and M_y are given and we want to extract M_x, it is similar to say that $m_1(i)$ and $m_y(i)$ are given and we want to extract $m_x(i)$. We are interested in the solution of the above modular equation if $m_x(i)$ is not known. To extract $m_x(i)$, we have two cases presented in Proposition 3.1.

Proposition 3.1. *If $m_1(i)$ and n are relatively primes, then $m_x(i)$ has a unique solution. Contrary to this, if $m_1(i)$ and n are not relative prime then there exists p solutions if $m_1(i)$ is multiple of p, or q solutions if $m_1(i)$ is multiple of q.*

Proof 3.1. *The first part of the proposition is trivial, we explain the second part i.e. the case where $m_1(i)$ and n are not relatively primes. In this case, the only common divisors possible to $m_1(i)$ and n, are p and q. That is, $m_1(i)$ is multiple of p or q. Suppose that $m_1(i)$ is multiple of p, then $m_1(i) = k \times p$, where $k \in \{1, \ldots, q-1\}$. Thus, p divides $m_1(i)$ and n, and from the* Eq. (20)*p also divides $m_y(i)$. We can then write $m_y(i) = p \times \tilde{m}_y(i)$,* Eq. (20)*signifies that there exists an integer l such that:*

$$k \times p \times m_x(i) = p \times \tilde{m}_y(i) + l \times p \times q, \tag{21}$$

and thus dividing by p gives:

$$k \times m_x(i) = \tilde{m}_y(i) + l \times q. \tag{22}$$

Thus, we have:

$$k \times m_x(i) \equiv \tilde{m}_y(i) mod\ \ q. \tag{23}$$

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

$$m_x(i) \equiv k^{-1} \times \tilde{m}_y(i) mod\ \ q. \tag{24}$$

This single solution modulo q leads to p solutions modulo n for the block $m_x(i)$: one before q, one between q and $2q$ and so on; in the case of $m_1(i)$ is multiple of q, we have in the same way q solutions. □

Since we may have multiple solutions for these noisy blocks of M_x, indeed a lot of values would be factors of the initial primes p and q, so they would give multiple solutions for each noisy block of M_x, these solutions must be less than or equal to $\{1, \ldots, q\}$ and the original value of the noisy block of M_x belongs to this solution set. We would keep storing this solution set for each noisy block.

In order to select the best value from the solution set for the noisy block and to remove the noisy blocks from the extracted M_x, we can take advantage of the homogeneity of the visual data, as there is high degree of coherence in the neighborhood of a given image pixel. So we calculate the average of the non-noisy pixel neighbors of noisy blocks of M_x and this average value is compared with each value of the solution set for the corresponding pixels, and then select the value from the solution set which is giving us the least distance from average value.

3.5.2. Probability distribution of noisy pixels

By using RSA cryptosystem, noisy pixel blocks are generated during the extraction of the original secret image. These noisy pixels are the values of the pixels of the original image which are factors of the initial primes being selected. Finding the probability distribution of these noisy pixels would help us in selecting the size of the block.

If p and q represent the initial primes in bits, then the probability of non-noisy or correct pixels is given by:

$$P(Non\_Noisy) = 1 - \left(\frac{1}{p} + \frac{1}{q}\right). \tag{25}$$

Obviously the total number of blocks (TB) is the sum of the total number of correct blocks (TCB) and noisy blocks (TNB), *i.e.* $TB = TCB + TNB$. If we group the pixels into a block the TCB would be:

$$TCB = TB \times \left(1 - \left(\frac{1}{p} + \frac{1}{q}\right)\right), \tag{26}$$

and the TNB would be given by:

$$TNB = TB \times \left(\frac{1}{p} + \frac{1}{q}\right). \tag{27}$$

The TNB will depend on the p and q values. Theoretically, to minimize TNB, we should have $p = q = \sqrt{n}$ but experimentally, that is not possible.

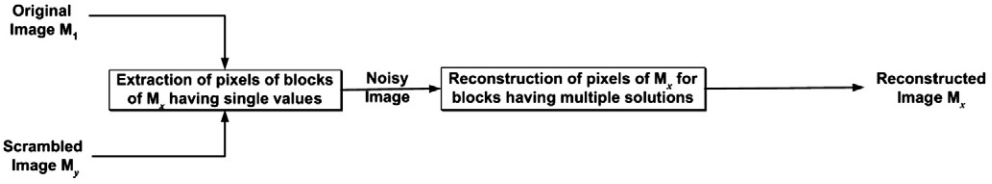

Fig. 5. Extraction and reconstruction of image M_x having pixels of image M_1.

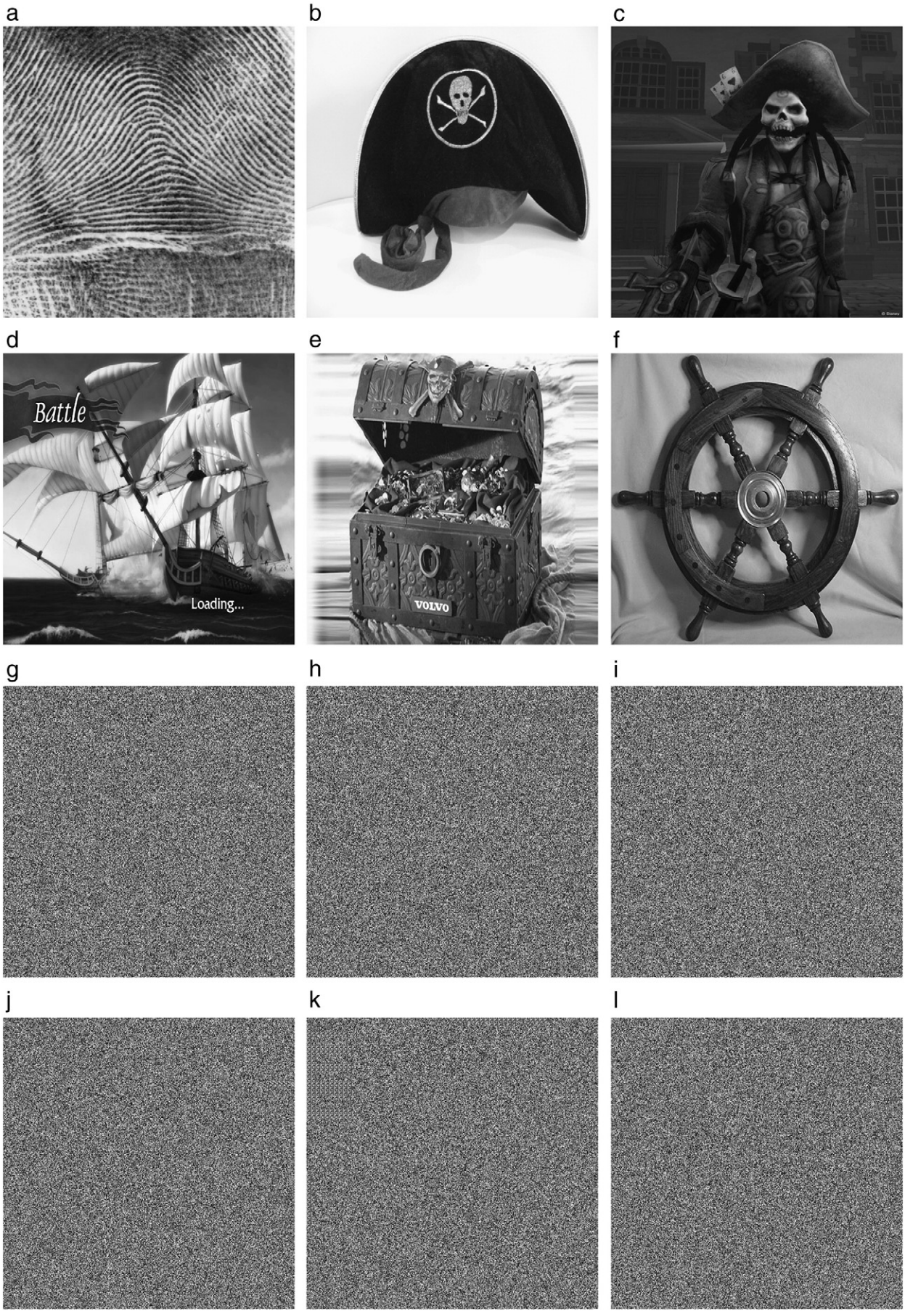

Fig. 6. a)–f) 6 secret original images $M_1, \ldots, M_l$, g)–l) Encrypted images with Paillier of (a)–(f), respectively.

Assume that we have a block of single pixel, then the value of p and q would be $p \approx 2^4$ and $q \approx 2^4$, if we assume the image size $512 \times 512 = 2^{18}$ pixels. Then $P(Non\_Noisy) = 1 - \left(\frac{1}{2^4} + \frac{1}{2^4}\right) = 1 - \frac{1}{2^3}$, and total number of correct blocks are given by $TCB = 2^{18} \times \left(1 - \left(\frac{1}{2^4} + \frac{1}{2^4}\right)\right) = 2^{18} - 2^{15}$ blocks. The total number of noisy blocks is given by $TNB = 2^{18} \times \left(\frac{1}{2^4} + \frac{1}{2^4}\right) = 2^{15} = 32768$ blocks.

If the block size is two pixels, then the number of blocks in the image will be 2^{17} and probability distribution of noisy blocks is $P(Non\_Noisy) = 1 - \left(\frac{1}{2^8} + \frac{1}{2^8}\right) = 1 - \frac{1}{2^7}$ and total number of correct blocks is given by $TCB = 2^{17}\left(1 - \left(\frac{1}{2^8} + \frac{1}{2^8}\right)\right) = 2^{17} - 2^{10}$ blocks, and total number of noisy blocks is given by $TNB = 2^{17} \times \left(\frac{1}{2^8} + \frac{1}{2^8}\right) = 2^{10} = 1024$ blocks.

The above theoretical results are based on the symmetrical selection of initial primes with respect to the number of bits. The term symmetrical means that the sizes of the primes are approximately equivalent. The selection of non-symmetrical primes is possible and mostly useful for the sake of security as the interest is only in the product of primes $p \times q = n$.

4. Experimental results and discussions

In this section, first experiments over the proposed approach on Paillier and RSA cryptosystems are discussed. Thereafter, security analyses for the proposed approaches are carried out in the shape of various tests like analysis of entropy, local standard deviation, correlation of adjacent pixels and key sensitivity test.

4.1. A full example of the proposed scheme involving Paillier Cryptosystem

For the demonstration of the proposed scheme over Paillier Cryptosystem, experimental tests were carried out on six gray level key-images and a secret map image, where each image is 8 bits/pixel having a size of 512×512 pixels.

Since the size of keys for encryption and decryption is chosen to be 512 bits, the block size is $n = 512/8 = 64$ pixels, so each block consists of 8×8 pixels. The encryption of six key-images M_1, …, M_6 and a secret map image M_x is given by C_1, …, C_6 and C_x. These encrypted images are further scrambled by applying a multiplication modulo n^2 in a specified order to get a new encrypted image C_y. C_y is then decrypted to get M_y which is our intended scrambled image to be shared through some transmission channel, not necessarily secured. Fig. 6.a–f presents the 6 original key-images of the players. Fig. 6.g–l illustrates the corresponding encrypted images obtained from the 6 key-images, where the image in Fig. 7.a is the secret image to be shared securely by the dealer (treasure map), Fig. 7.b is the encrypted version of Fig. 7.a while Fig. 7.c corresponds to the encrypted image from the multiplication of the 7 encrypted images from Figs. 6.g–l and 7.b.

4.1.1. Extraction with the l key-images

In Fig. 8, we show the decryption and extraction of the shared secret image. Fig. 8.a illustrates the decrypted image which is actually the sum of the $l+1$ images that is used for the onward transfer. Finally, Fig. 8.b shows the reconstructed image which is 100% same as the original image M_x. It may be noted over here that the reconstructed image does not depend on the order to subtract the l key-images.

4.1.2. A model of generalized (l-1,l)

In order to understand the tolerance of the proposed scheme in the absence or modification of a key-image, it has been observed that the proposed scheme under Paillier cryptosystem is tolerably sensitive to any change in the key-image. This tolerance reflects the additive homomorphic property of Paillier. Experiments were performed in order to understand the behavior of additive homomorphic property while making changes in any of the key-image and observe its impact over the resultant extracted image. For this purpose, during the extraction, one of the key-images from Fig. 6.a–f was replaced with a

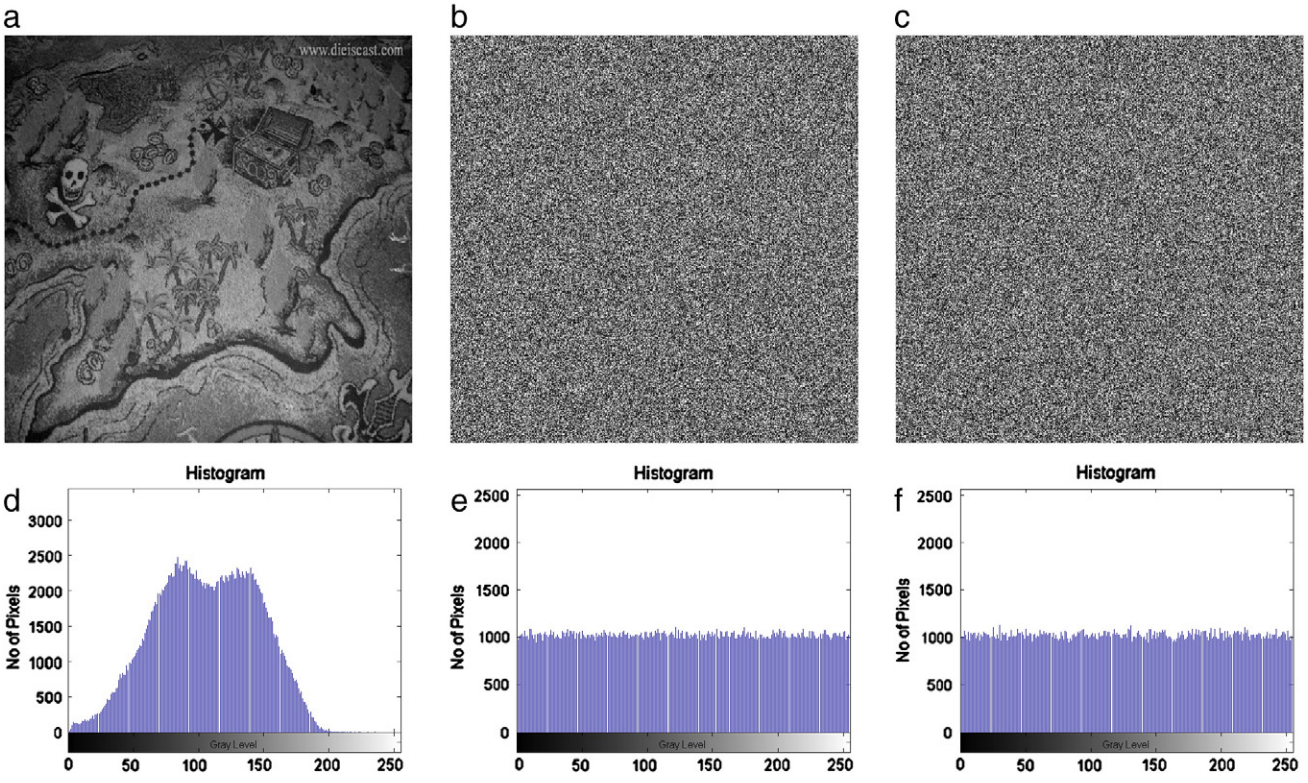

Fig. 7. a) Secret image to be shared (M_x), b) Encrypted image with Paillier of (a), c) Encrypted image C_y obtained from multiplication of Fig. 6.g to 6.l and Fig. 7.b, d), e) and f) Histogram of (a), (b) and (c), respectively.

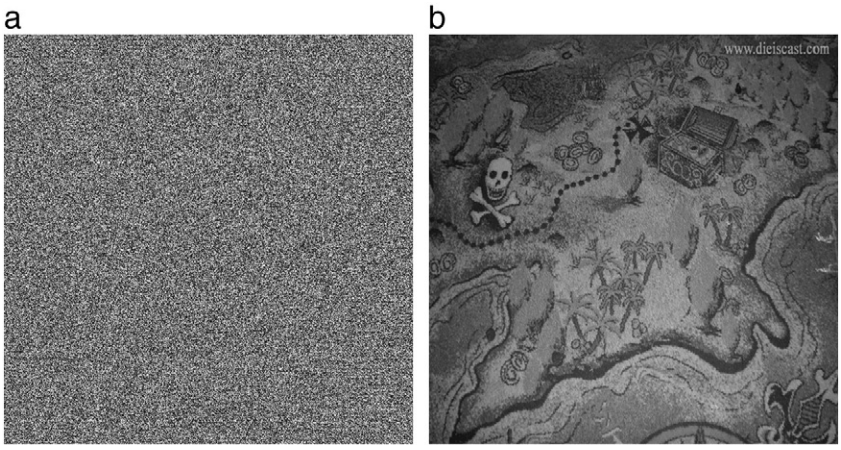

Fig. 8. a) Decrypted image of Fig. 7.c which is the scrambled shared image, b) Extracted image.

JPEG compressed image at different compression quality factors. Fig. 9.b and d represent the extracted secret images when the key-image in Fig. 6.d is compressed with JPEG at a quality factor of 25%, as shown Fig. 9.a, and 50%, as shown Fig. 9.c. Sharp observation reveals some noisy pixels in the extracted images illustrated Fig. 9.b and d, but the overall quality is good, as corraborated by its PSNR which was observed to be greater than 35 dB. It can be concluded from these experiments that small variations in the original data can be tolerable and the extraction of the secret image is possible.

To make our scheme analogous to the (k, l) threshold scheme, described in Section 2.6, we have tailored our method to a $(l-1, l)$ scheme by replacing one of the key-images with a neutral homogeneous image where all the pixels are set to 128 intensity, as shown in Fig. 10.a. This replacement is tantamount to the absence of the image in question, making our scheme conform to the (k, l) threshold. A number of experiments were performed to this effect, *e.g.*Fig. 10.b. illustrates the extracted image when the key-image presented Fig. 6.b. is missing and replaced by a homogeneous image given in Fig. 10.a.

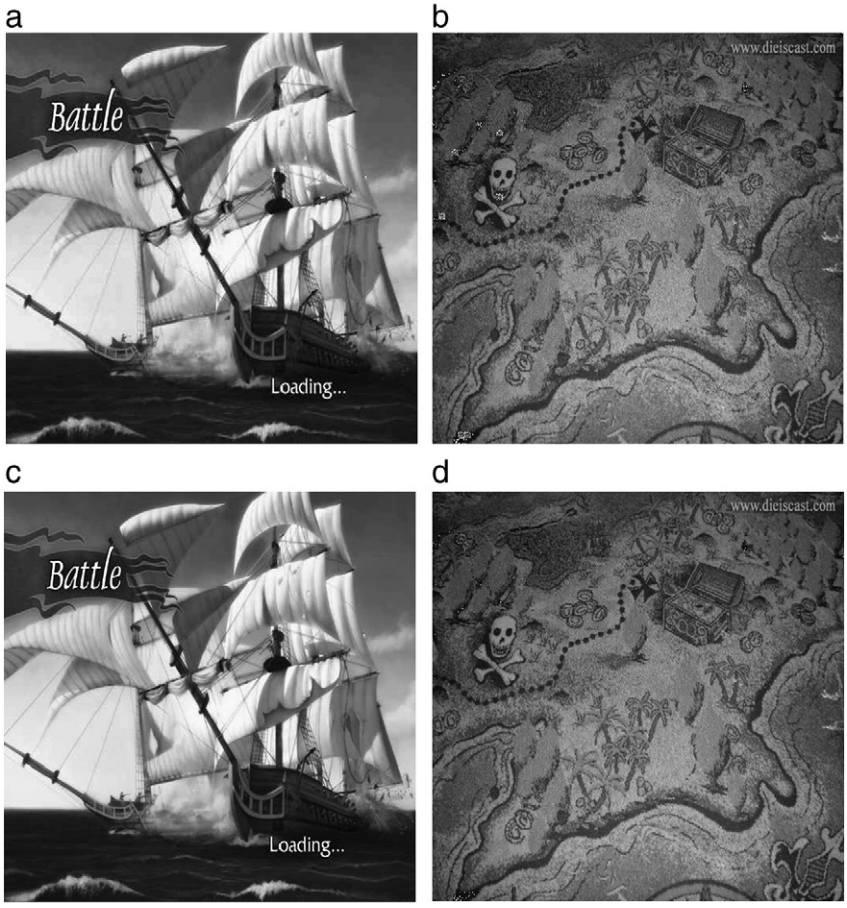

Fig. 9. a) Compressed key-image at a quality factor of 25% (PSNR = 36.58 dB), b) Extracted secret image using the compressed image of Fig. 9.a (PSNR = 36.48 dB), c) Compressed Image at a quality factor of 50% (PSNR = 38.90 dB), d) Extracted secret image using the compressed image of Fig. 9.c (PSNR = 38.78 dB).

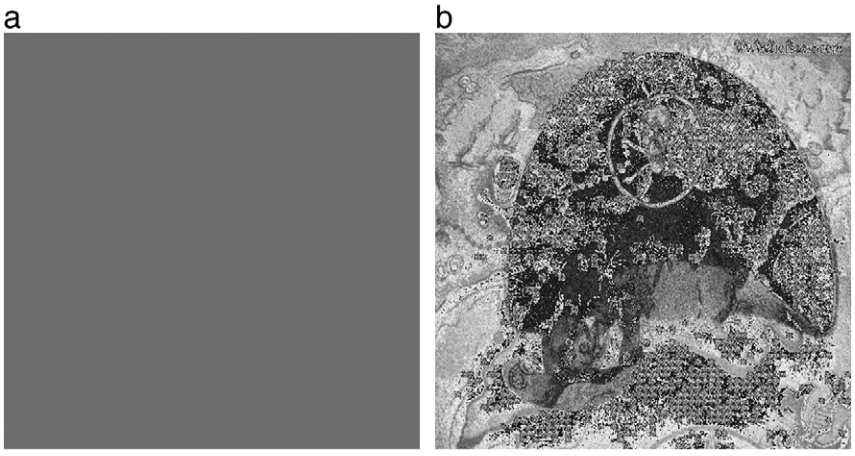

Fig. 10. a) A homogeneous image, b) The resultant extracted secret image.

This experimentation shows that the extraction of the secret image is approximately possible in the absence of one key-image.

4.2. A full example employing RSA

Fig. 11 illustrates an example of the proposed method using the RSA algorithm. Like the Paillier cryptosystem, experimental tests were carried out on six gray level key-images and a secret Lena image, where each image is 8 bits/pixel having a size of 512×512 pixels. The size of the keys for encryption and decryption is chosen to be 512 bits, thus the block size becomes $64=8\times8$ pixels. With this key size, the probability of pixel block values which are factors of prime product is very small as explained in Section 4.2.1. Fig. 11.a–f presents the original images and Fig. 11.g–l presents the corresponding encrypted images with the RSA. Fig. 12.a presents the secret Lena image to be shared among the participants, Fig. 12.b presents the encrypted Lena image, Fig. 12.c presents the multiplication of the encrypted Lena image with the other encrypted images using the proposed protocol. Fig. 12.d–f presents the histogram of the images in Fig. 12.a–c respectively. Fig. 13.a presents the decrypted image of Fig. 12.c, while Fig. 13.b presents the extracted secret image of Lena.

As explained in Proposition 3.1, there may exist pixel block values which are factors of the prime product and which may lead to multiple solutions while extracting the secret image. For this purpose, we used Lena image and another image keeping the prime values too small to have multiple solutions. In Fig. 14, we show the extraction and reconstruction of the shared secret image. Fig. 14.a illustrates the extracted image with noisy blocks of two pixels having multiple possible solutions.[2] Finally, Fig. 14.b shows the reconstructed image, where the missing or noisy pixels were reassigned the mean value of non-noisy neighboring pixels. This latter image is visually more close to the original image, as revealed by its PSNR of 47.8 *dB* with reference to the original Lena image. This value shows high degree of likeness between the original and the reconstructed image.

4.2.1. Selection of block size

Experiments for the selection of block size have been performed on 100 different images, based on the selection of primes of nearly equal sizes. We found that the larger the block size, greater will be the security and lower the risk of the noisy pixels. Table 1 shows the probability distribution of noisy blocks for various block sizes. For our experiments, the initially selected primes were symmetric with respect to bit size. In Table 1, we can see that as the block size increases, the probability of noisy blocks decreases. The experimental results over an average of 100 images with noisy blocks (third column) confirms the statement about the inverse relationship between the size of the block and the number of noisy blocks. We observe that, as a rule of thumb, when the block size increases beyond 4 pixels, noisy pixel blocks are rarely observed.

For the sake of simplicity, note that primes of nearly equal sizes have been taken for these experiments. Fig. 15.a–d shows the graphical representation of the results when single, double, $2\times2=4$ and $4\times4=16$ pixel block are selected.

4.3. Security analysis

4.3.1. Analysis of entropy and local standard deviation

The security of the encrypted images can be measured by considering the variations (local or global) 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 distortion that is mostly used to characterize the texture in the input images. If an image has 2^k gray levels α_i with $0\le i<2^k$ 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:

$$H(X)=-\sum_{i=0}^{2^k-1}P(\alpha_i)log_2(P(\alpha_i)). \tag{28}$$

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 an image, the information redundancy r is defined as:

$$r=k-H(X). \tag{29}$$

When r is close to 0, the security level is acceptable. Theoretically an image is an order-M Markov source, with M being the image size. In order to reduce the complexity, the image is cut into small blocks 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 $0\le i<2^{k^n}$ and the order-n entropy $H(X')$ is defined as:

$$H(X')=H(X^n)=-\sum_{i=0}^{2^{k^n}-1}P(\beta_i)log_2(P(\beta_i)). \tag{30}$$

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

[2] For visual illustration we have chosen blocks of two pixels even if the security level is not high.

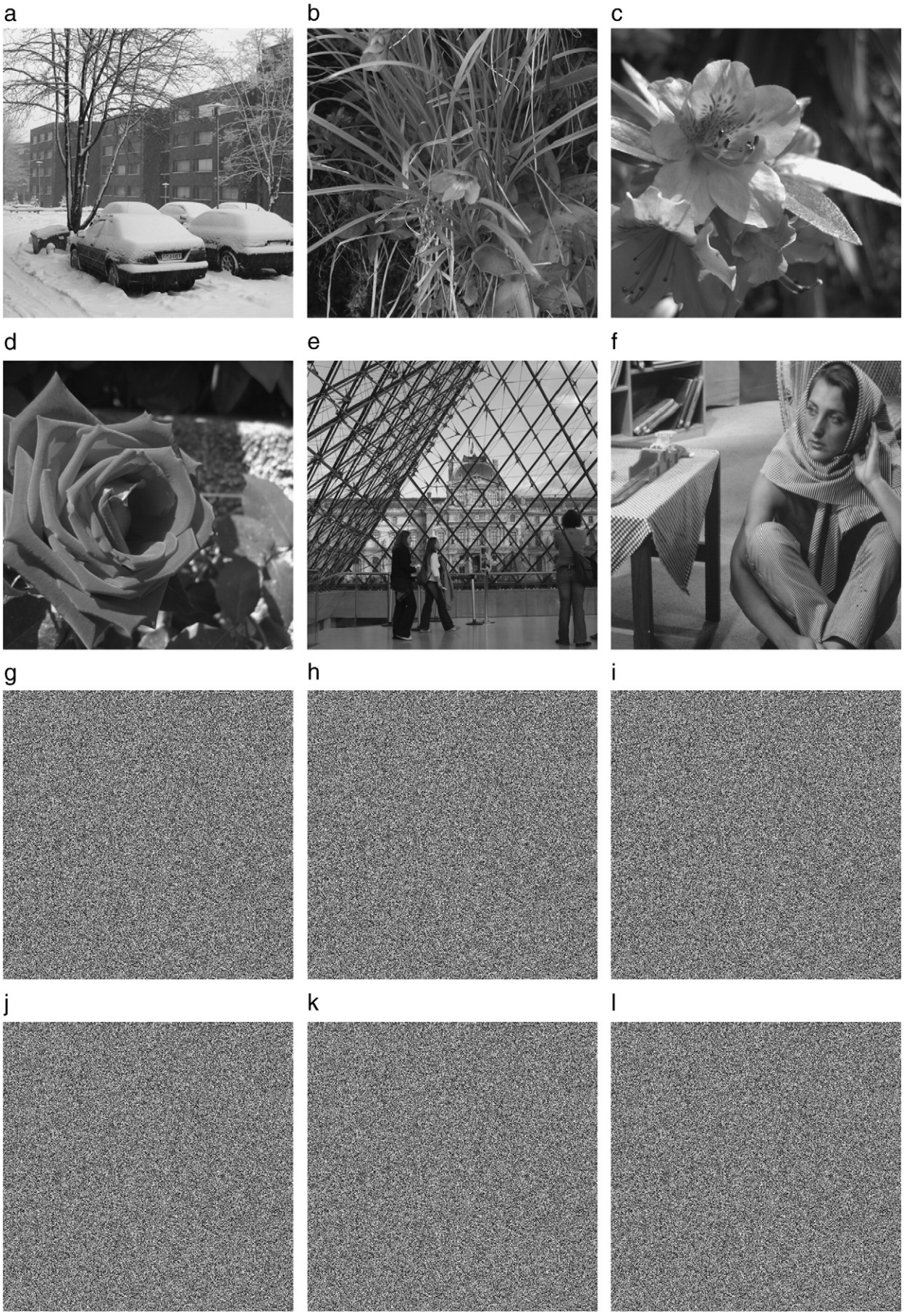

Fig. 11. a–f) 6 key-images M_1, …, M_l, g–l) Encrypted images with RSA of (a)–(f), respectively.

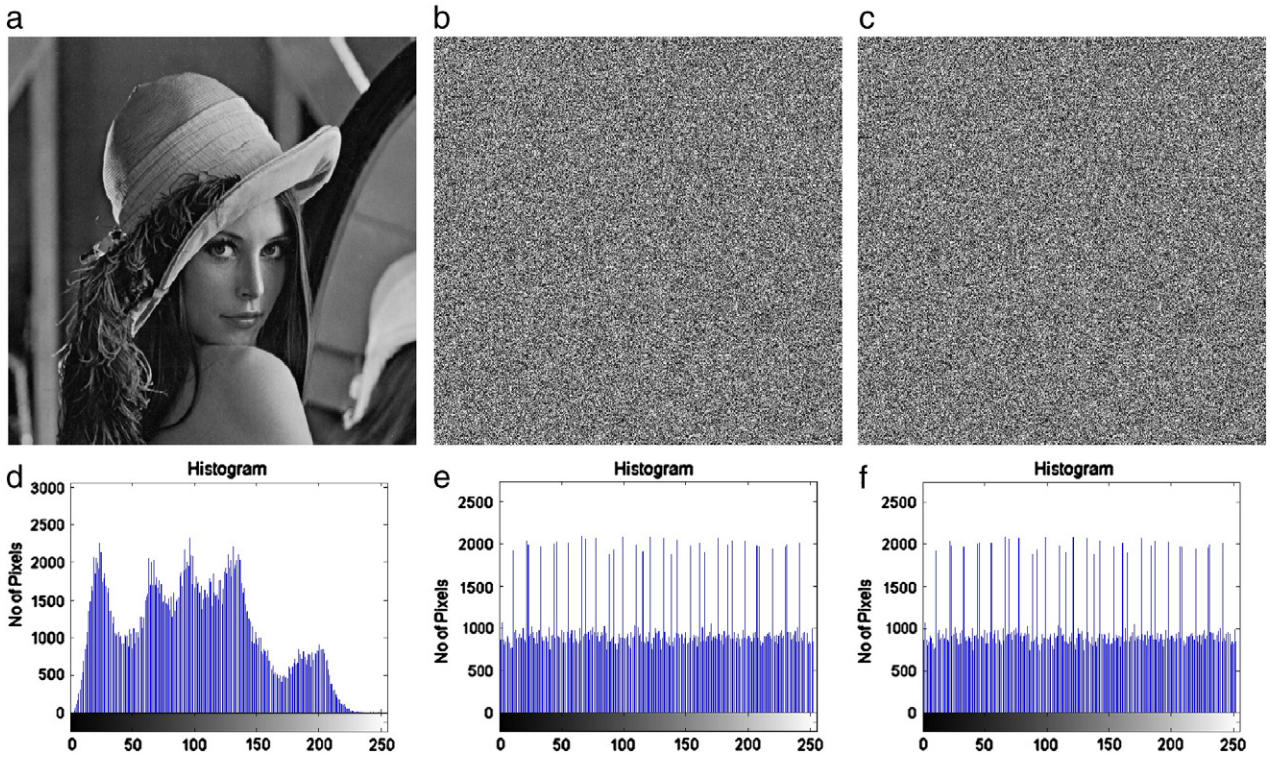

Fig. 12. a) Secret image to be shared (M_x), b) Encrypted image with of (a), c) Encrypted image C_y obtained from multiplication of Fig. 11.g to Figs. 11.l and 12.b, d), e) and f) Histogram of (a), (b) and (c), respectively.

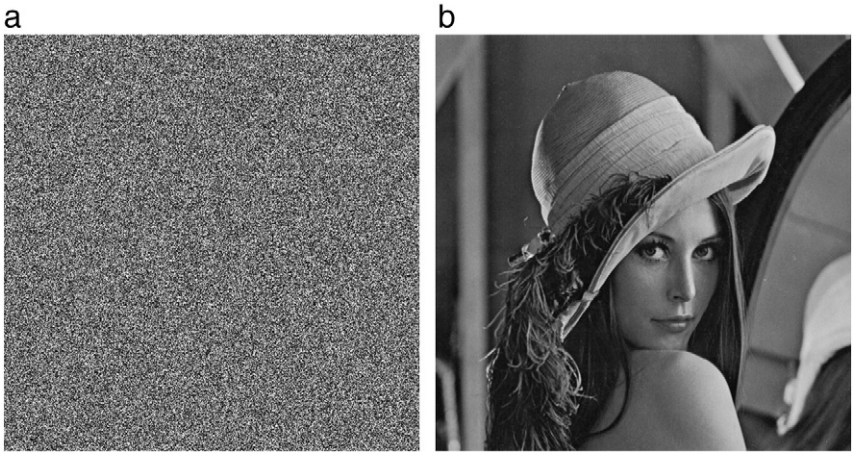

Fig. 13. a) Decrypted image of Fig. 12.c which is the scrambled image, b) Extracted image.

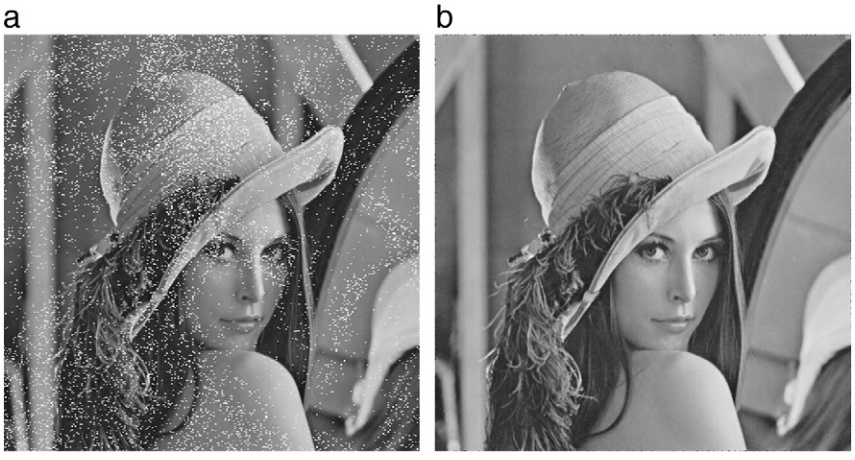

Fig. 14. a) Extracted image with blocks of two pixels, b) Reconstructed image.

Table 1
Probability Distribution of noisy blocks for image size of **512×512.**

Block size in pixels	Number of blocks	Average theoretical noisy pixels blocks	Average experimental noisy pixels blocks
1	2^{18}	2^{15}	34,650
2	2^{17}	2^{10}	1223
2×2=4	2^{16}	2	1.8
4×4=16	2^{14}	2^{-53}	0
8×8=64	2^{12}	2^{-243}	0
16×16=256	2^{10}	2^{-1018}	0
32×32=1024	2^{8}	2^{-4091}	0
64×64=4086	2^{6}	2^{-16380}	0

Table 2
1st order and 2nd order entropy of original images over Paillier scheme and RSA scheme.

	H(X) 1st order entropy (bits/pixels)	2nd order entropy (bits/pixels)
Original map image	7.28	13.35
Paillier Scheme	7.99	15.80
Original lena image	7.45	12.33
RSA scheme	7.99	15.76

minimum redundancy *i.e.* $r \approx 0$, as required by Eq. (29), we should have k=8 bits/pixel for Eq. (28) and k=16 or 24 bits/block for Eq. (30).

We also analyzed the variation of the local standard deviation $\sigma(j)$ for each pixel $p(j)$ by taking into account its m neighbors to calculate the local mean $\overline{p(j)}$ according to the following equation:

$$\sigma(j) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(p(i)-\overline{p(j)}\right)^2}, \tag{31}$$

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

Fig. 16 shows the histograms of the original and the scrambled images using the proposed two schemes based on the Paillier and RSA cryptosystems. Here a visible change in the histogram shape can be observed between the plain images and the corresponding transmitted image. In Fig. 16.d and h, we can see uniform distributions of the gray level values among the pixel coordinates of the transmitted image, while for the original images, Fig. 16.c and g, their peaks of gray level values which signify some shapes or objects are present.

From Eq. (28) we get high entropy values of 1st order and 2nd order over the transmitted scrambled images, while using the proposed schemes for Paillier and RSA. The information redundancy r for the scrambled image, involving the Paillier scheme over 1st order entropy is 0.01 and that for 2nd order entropy is 0.20 which are negligible as compared to those for the original image, *i.e.* 0.72 and 2.65 for 1st and 2nd order respectively (Table 2). The value of r for scrambled image, involving RSA-based scheme, over 1st order entropy is 0.00079 and over 2nd order entropy is 0.24 as against 0.55 and 3.67 for 1st and 2nd order respectively (Table 2), with the original image.

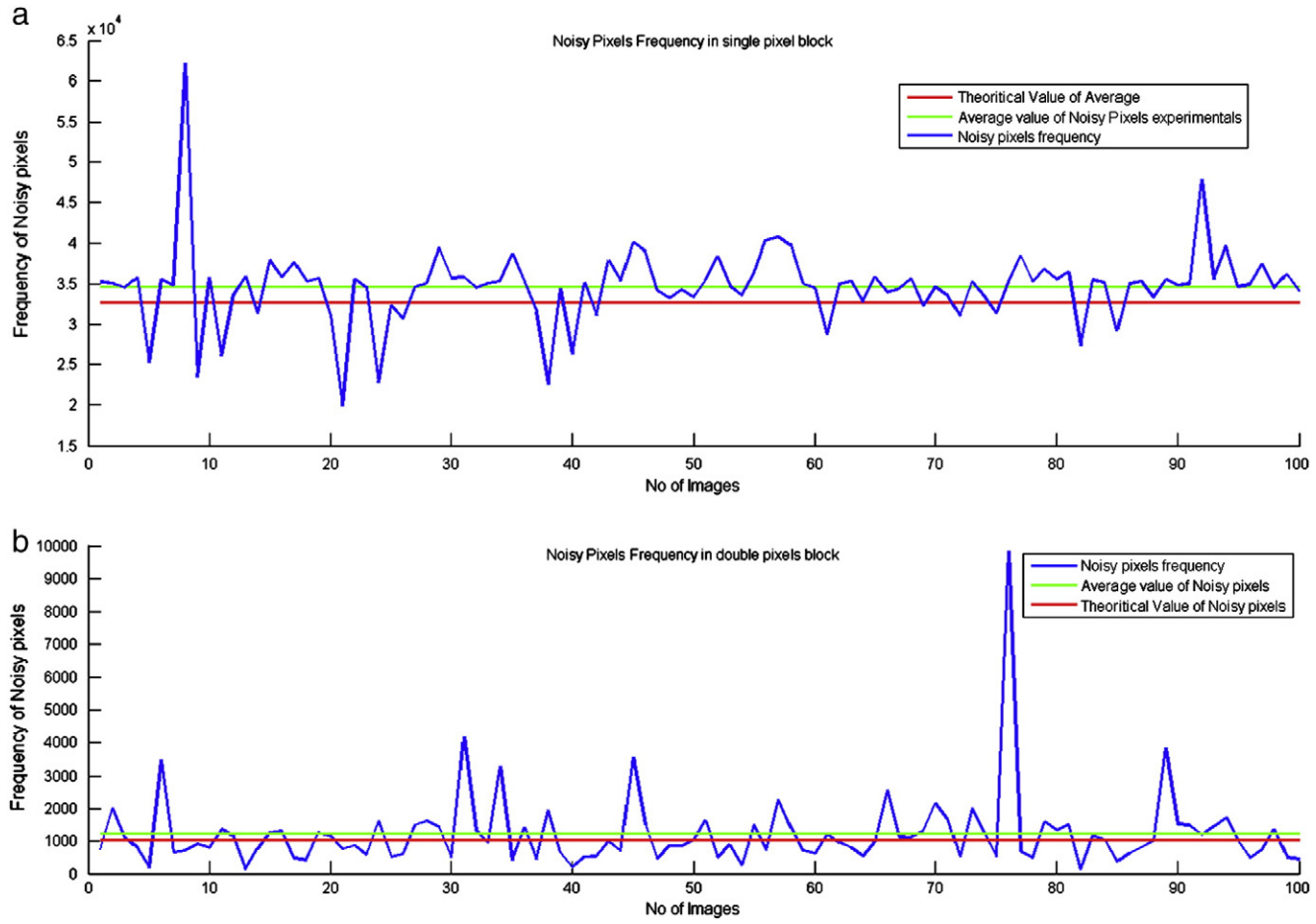

Fig. 15. Noise frequency for 100 images of size 512×512: a) In single pixel block, b) In 2 pixel block, c) In 4 pixel block, d) In 16 pixel block.

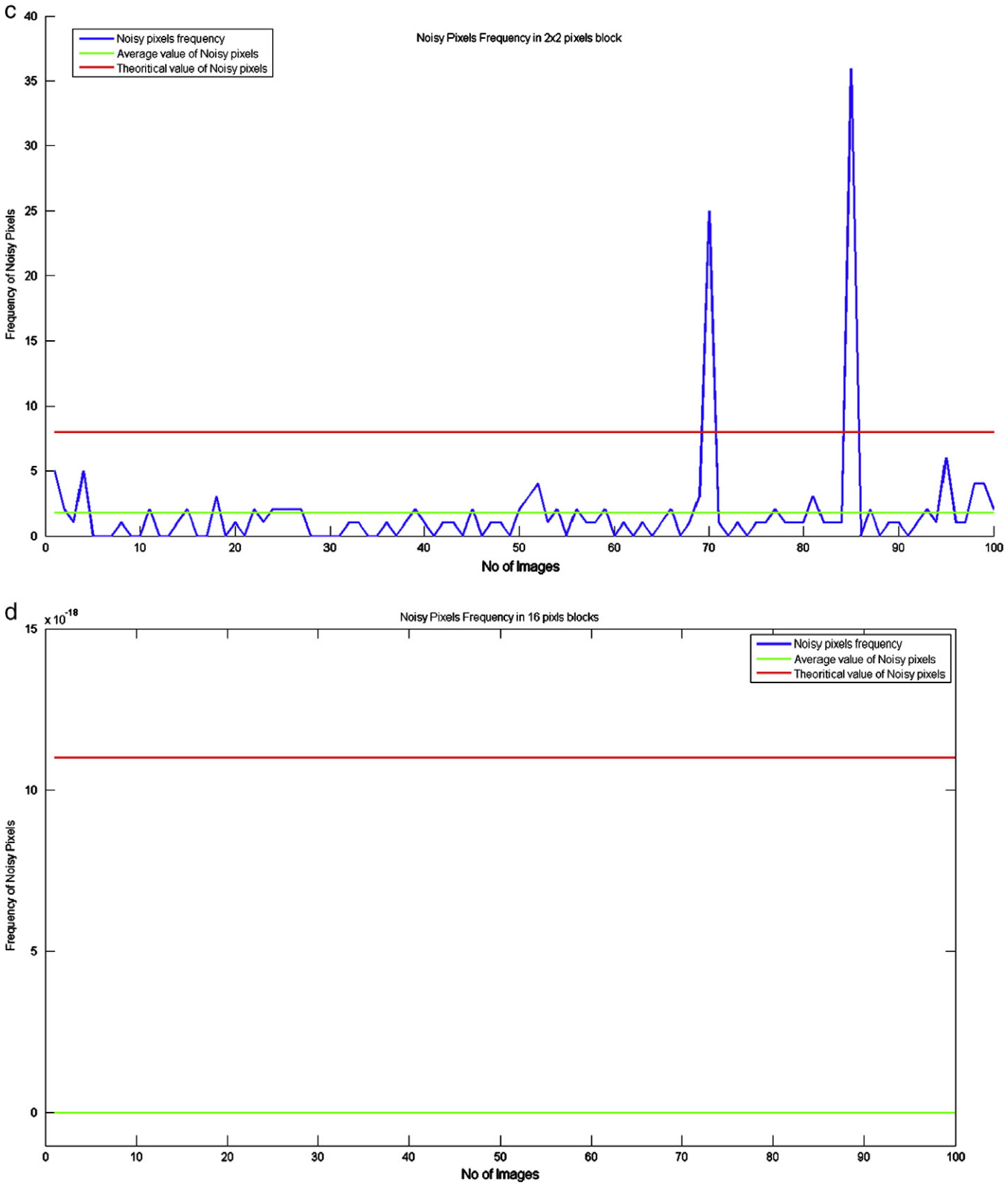

Fig. 15 *(continued)*.

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 73.86 gray levels for the scrambled image of Fig. 16.b using Paillier-based scheme, whereas the mean local standard deviation equals 13.38 gray levels for the original Map image. Similarly, the mean local standard deviation equals 72.60 gray levels for the scrambled image of Fig. 13.a using RSA-based scheme, whereas the mean local standard deviation equals 6.21 gray levels for the original Lena image. These analyses show that the scrambled images are protected against statistical attacks.

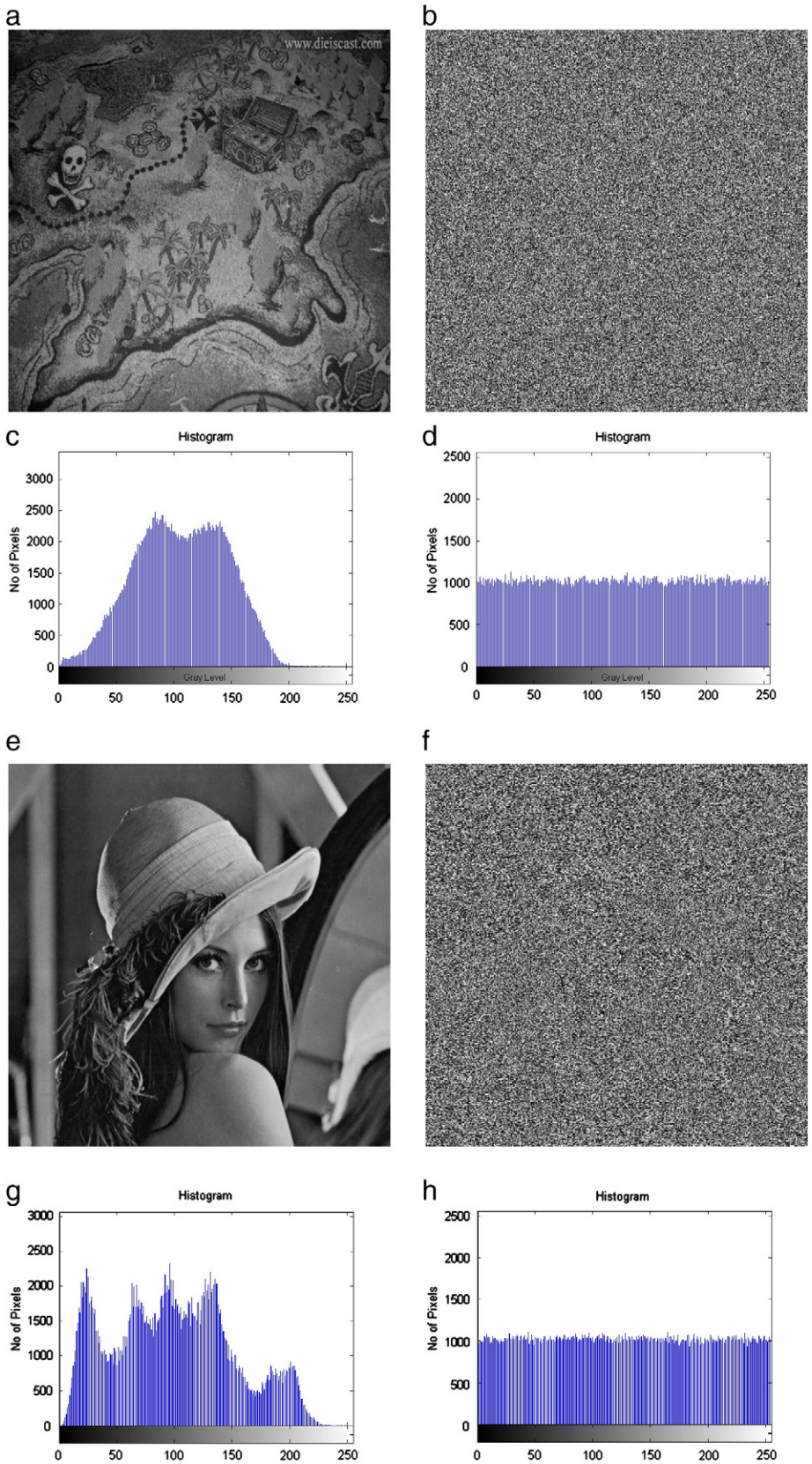

Fig. 16. a) Original map image, b) Scrambled image from Fig. 8.a for safe transmission of (a) using Paillier, c, d) Histogram of (a) and (b) respectively, e) Original Lena image, f) Scrambled image from Fig. 13.a for safe transmission of (e) using RSA, g, h) Histogram of (e) and (f) respectively.

4.3.2. Correlation of adjacent pixels

Visual data is usually highly correlated *i.e.* pixel values are highly probable to repeat in horizontal, vertical and diagonal directions. Since RSA public-key cryptosystem is not random in nature, it gives the 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

Table 3
Correlation of horizontal, vertical and diagonal adjacent pixels in two images using Paillier cryptosystem.

	Plain map image	Scrambled image
Horizontal	0.8287	1.7112×10^{-4}
Vertical	0.8529	0.0027
Diagonal	0.7905	0.0034

Table 4
Correlation of horizontal, vertical and diagonal adjacent pixels in two images using RSA cryptosystem.

	Plain Lena image	Scrambled image
Horizontal	0.9936	0.00240
Vertical	0.9731	0.00013
Diagonal	0.9309	0.00022

public-key encryption will produce the same results, and a cryptanalyst may have enough clues to the information content related to the original image. An image based cryptosystem is considered robust against statistical attacks if it succeeds in providing low correlation between the neighboring pixels. The proposed encryption scheme generates a ciphered image with low correlation among the adjacent pixels. A correlation of a pixel with its neighboring pixel is given by ordered pair (x_i, y_i) where y_i is the adjacent pixel of x_i:

$$corr_{(x,y)} = \frac{1}{M-1}\sum_{0}^{n}\left(\frac{x_i-\overline{x_i}}{\sigma_x}\right)\left(\frac{y_i-\overline{y_i}}{\sigma_y}\right), \tag{32}$$

where M represents the total number of tuples (x_i, y_i), $\overline{x_i}$ and $\overline{y_i}$ represents their respective means and σ_x and σ_y represent their respective standard deviations.

In Table 3, we can see correlation values of original map image and the transmitted scrambled image using Paillier cryptosystem. Here the correlation coefficients in horizontal, vertical and diagonal directions are higher in plain map image while these values are very small in the encrypted images. Similarly, in Table 4, we can see correlation values of Lena image and the corresponding transmitted scrambled image using RSA cryptosystem. It can be noticed from these tables that the proposed schemes minimize the correlation coefficients in horizontal, vertical and diagonal directions.

4.3.3. Key sensitivity test

Robustness of a cryptosystem can be improved if it is made highly sensitive towards the key. The more sensitive the visual data to the key, more would be the data randomness higher the value of the entropy, leading and hence to lower visual correlation among the pixels of the image. For this purpose, a key sensitivity test was conceived where we picked one key, K_1, of 512 bits and then apply the proposed techniques for encryption, followed by a one bit change in the length of the key *i.e.* K_2 of 511 bits and then re-apply the proposed encryption techniques. Numerical results show that the proposed approach is highly sensitive towards the key change *i.e.* a totally different version of scrambled image was produced when the keys were changed, as shown in Fig. 17, which is an example of using RSA-based scheme. The correlation between the two encrypted images which are produced by applying the proposed approach, with two

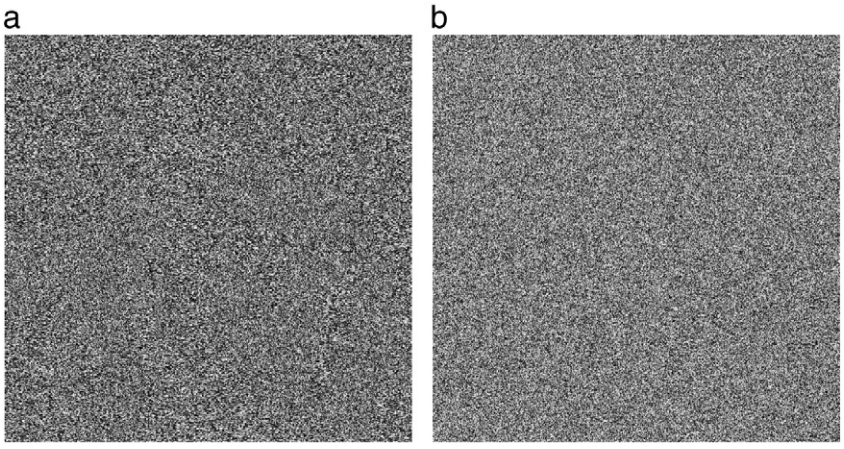

Fig. 17. Key sensitivity test: a) Encrypted image with key, K1, b) Encrypted image with key, K2.

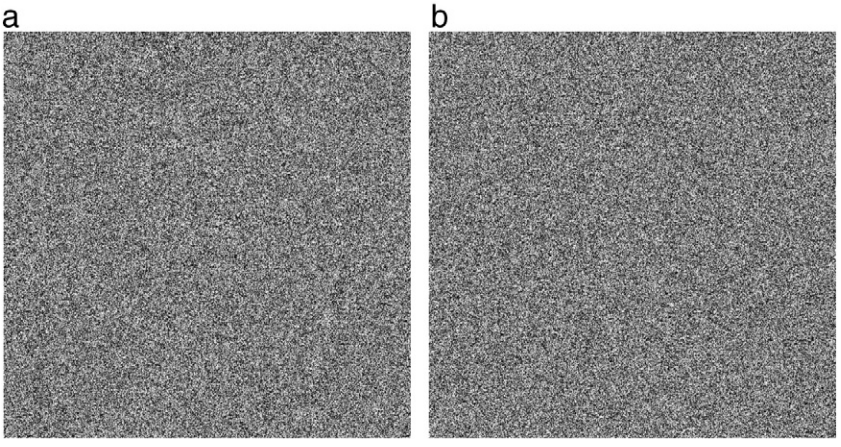

Fig. 18. a) Image encrypted with K_1 and decrypted with K_2, b) Reconstructed image.

different keys K_1 and K_2, is given by 0.118 for RSA-based scheme and 0.0952 for Paillier-based scheme, which means there is negligible amount of correlation among the pixels of the ciphered images produced with different keys. Also, if we encrypt an image with one key (K_1) and decrypt with another key (K_2) and then apply the proposed scheme for the reconstruction of the original image, we cannot get the original image. This observation over the RSA-based scheme can be deduced from Fig. 18.

5. Conclusions

In this paper, we proposed a method for sharing a secret image using key-images by exploiting the additive homomorphic property of Paillier cryptosystem and the multiplicative homomorphic property of RSA cryptosystem. The proposed method is analogous to the VSS scheme but here we took a number of different key-images and then exploited the homomorphic property of public key cryptosystems to devise a protocol where a scrambled secret image is formed. Of the two schemes employed, *i.e.* Paillier and RSA, it has been shown that the extracted image from additive homomorphic Paillier cryptosystem observes no change. On the other hand, in the application of the RSA, which is multiplicative homomorphic, care must be taken while selecting the size of the pixel blocks for encryption, in order to avoid prime multiplicity problem. We showed that the extraction of the original secret image, involving the RSA algorithm, is possible if a block size of more than 4×4 is used whereby the probability of noisy pixels approaches 0. The advantages of the proposed approach are the low computational cost at the extraction process, limiting of the dealer's intervention in the secret sharing process and no increase in the size of shared scrambled image. We can use the proposed scheme in various applications on any public key cryptosystem that satisfies the desired additive or multiplicative homomorphic property.

References

[1] B. Schneier, Applied Cryptography, Wiley, New-York, USA, 1995.
[2] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, Morgan Kaufmann, 2008.
[3] D. Rappe, Homomorphic Cryptosystems and their Applications, Cryptology ePrint Archive, Report 2006/001, 2006.
[4] M. Kuribayashi, H. Tanaka, IEEE Transactions on Image Processing 14 (12) (2005) 2129.
[5] M.V. Droogenbroeck, R. Benedett, Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium, Sept. 2002, p. 90.
[6] A. Alattar, G. Al-Regib, S. Al-Semari, Proc. IEEE Int. Conf. on Image Processing, vol. 4, 1999, p. 256.
[7] H. Cheng, X. Li, IEEE Transactions on Signal Processing 48 (8) (Aug. 2000) 2439.
[8] A. Shamir, Communications of the ACM 22 (11) (1979) 612.
[9] M. Naor, A. Shamir, Visual Cryptography, Springer-Verlag, 1995.
[10] P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residousity Classes, 1592, Springer-Verlag, 1999, p. 223.
[11] A. Uhl, A. Pommer, Image and Video Encryption: From Digital Rights Management to Secured Personal Communication, Springer, 2005.
[12] E. Gamal, IEEE Transactions on Information Theory (1985) 469.
[13] C. Fontaine, F. Galand, EURASIP Journal on Information Security 2007 (1) (2007) 1.
[14] J. Borie, W. Puech, M. Dumas, ICDIA 2002, Diagnostic Imaging and Analysis, Shanghai, R.P. China, Aug. 2002, p. 250.
[15] C. Fu, Z.-C. Zhang, Y. Chen, X.-W. Wang, An Improved Chaos-Based Image Encryption Scheme, ICCS '07: Proceedings of the 7th international conference on Computational Science, Part I, 2007, p. 575.
[16] J. Hu, F. Han, Journal of Network and Computer Applications 32 (4) (2009) 788.
[17] J. Giesl, K. Vlcek, ICGST International Journal on Graphics, Vision and Image Processing, GVIP 09 (2009) 19.
[18] S.R.M. Prasanna, Y.V.S. Rao, A. Mitra, International Journal of Computer Vision 1 (2) (2006) 132.
[19] G. Ateniese, C. Blundo, A.D. Santis, D.R. Stinson, Visual cryptography for general access structures, Electronic Colloquium on Computational Complexity (ECCC) 3 (12) (1996).
[20] S. Droste, CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, London, UK, 1996, p. 401.
[21] Á.M. del Rey, G.R. Sánchez, IWANN (1), Vol. 5517 of Lecture Notes in Computer Science, Springer, 2009, p. 1200.
[22] M. Iwamoto, L. Wang, K. Yoneyama, N. Kunihiro, K. Ohta, IEICE Transactions 89-A (5) (2006) 1382.
[23] M. Iwamoto, H. Yamamoto, H. Ogawa, IEICE Transactions 90-A (1) (2007) 101.
[24] S. Cimato, R.D. Prisco, A.D. Santis, Theoretical Computer Science 374 (1–3) (2007) 261.