Multimedia Tools and Applications (2021) 80:29067–29093
https://doi.org/10.1007/s11042-021-11068-8

Multi-image steganography and authentication using crypto-stego techniques

Himani Sharma[1] · D. C. Mishra[2] · R. K. Sharma[2] · Naveen Kumar[1]

Received: 6 May 2020 / Revised: 2 May 2021 / Accepted: 11 May 2021 /
Published online: 17 June 2021

Abstract

It is a necessity to protect sensitive information in digital form from an adversary who may indulge in cyber-crimes such as modification, masquerading, and replaying of data. Security systems designed to counter such attacks must keep abreast of the adversary. In this paper, we have proposed a novel multi-image crypto-stego technique using Rabin cryptosystem and Arnold transform that provides a mechanism to hide digital data in the form of text, image, audio, and video. The proposed technique is a novel approach for (n,n) secret sharing that prevents attack by an intruder impersonating as a shareholder. In the proposed technique, the header information is created to retrieve data in the correct order. Randomized encrypted data and partial header information are camouflaged in the edges of multiple images in an adaptive manner. Minimal and distribution sequence keys distribute data in shares. Experimental results yield high values of PSNR and low values of MSE for the audio, image, video signals. Further, as the entropy values for original cover image coincide with the crypto-stego image up to the third place of decimal, the secret message will go unnoticed. Sensitivity analysis reveals that even a minor variation in a single share makes the recovery of the secret message infeasible. Comparison with the state of the art techniques indicates that the proposed technique either scores over its competitors or performs equally well in terms of standard evaluation metrics.

Keywords Encryption · Decryption · Arnold transform · Rabin cryptosystem · Multi-image steganography · Information security · Secret sharing

✉ Himani Sharma
himani.sharma.cs.du@gmail.com

D. C. Mishra
deepiitdelhi@gmail.com

R. K. Sharma
rksharma@maths.iitd.ac.in

Naveen Kumar
nk.cs.du@gmail.com

[1] Department of Computer Science, University of Delhi, Delhi, 110007, India

[2] Department of Mathematics, Indian Institute of Technology Delhi, Delhi, 110016, India

1 Introduction

Protecting sensitive information while sharing it among various stakeholders is an important aspect of information security. There are several techniques to protect the data. In cryptography, the message of interest is encrypted prior to communication across an insecure channel so that if the message is intercepted by an intruder during communication, it remains incomprehensible to him/her. Researchers have proposed various techniques for image encryption. Abuturab [4] proposed a scheme to decompose an image into three RGB channels and applied Discrete Cosine Transform (DCT) and gyrator transforms to secure the message. Several researchers have proposed modifications of DCT in various applications such as face and palm-print recognition [30, 31]. Hayat and Azam [19] presented a hybrid image encryption scheme based on substitution boxes and pseudo random number methods using elliptic curves. Luo et al. [38] proposed an improved chaotic algorithm for shuffling and substitution for encrypting images. Rabin [43] developed an asymmetric encryption scheme based on a variant of RSA using two as the exponent, thus making computations fast. Several schemes that were developed to correctly decipher roots for decryption, were vulnerable to chosen plaintext attack [9, 20, 39]. Many variants of the scheme were developed to address the issue of indistinguishability against chosen plaintext attack (IND-CPA) [27–29]. Recently, Rabin cryptosystem has been integrated with other techniques such as error correction codes [16] to provide resistance from forgery [15].

Steganography is a technique for evading detection by hiding secret digital messages in innocent looking multimedia files containing text, image, audio, or video, so that an intruder does not get any clue of a secret message being communicated. Main issues in steganography include capacity, security, and robustness [51]. Steganography may be applied in spatial, spectral, or adaptive domain. Researchers have worked on different aspects to improve the embedding capacity and security of data. Abdulla et al. [2] proposed a data hiding technique with improved stego-image quality by decomposing the pixel intensity value into 16 virtual bit planes suitable for embedding. Taking advantage of similarity between neighbouring pixels, Tang et al. [48] proposed a high capacity data hiding technique using multilayer embedding. Chen et al. [13] proposed a color image hiding scheme in which RGB components are first scrambled using Arnlod transform and subsequently hidden using Gerchberg–Saxton algorithm in fractional Fourier domain. Kim et al. [25] proposed a high capacity scheme using hamming code parity function to exploit embedding efficiency of encrypted halftoned image. By exploiting the bitmap coefficients of an image based on histogram modification, the authors proposed a lossless data hiding method for absolute moment block truncation coding compressed gray scale images [24]. Abdulla et al. [1, 3] proposed a technique to improve the security by Least Significant Bit Replacement (LSBR) method that increased the embedding efficiency by about 50%. They used a bit plane mapping technique instead of bit plane replacement, thus improving similarity between the secret image bit stream and the cover image.

Cryptography and steganography techniques, when combined, provide better security to the sensitive data [8, 41]. Enhanced imperceptibility is achieved by hiding secret data in edges of the cover image [7, 47]. Kim et al. [26] presented a novel data hiding method based on adaptive block truncation coding (BTC) that exploited edge quantization using optimal pixel adjustment process at the least significant bit (LSB) pixels to obtain a high image quality.

In this paper, we have proposed a multi-image authentication mechanism using crypto-stego techniques. Secret messages in the form of text, image, audio, or video are encrypted using Rabin cryptosystem and randomized using Arnold transform. Subsequently, the transformed messages are embedded in the edges of multiple images using adaptive LSBMR

technique [37]. The proposed technique is an (n,n) secret sharing technique that yields high quality shares. The proposed approach is suitable for sharing highly sensitive information where all the shareholders' information are necessary to retrieve the secret message. Due to its large storage capacity, it is also suitable for securing large data.

1.1 Road map of the paper

The rest of this paper is organized as follows: in Section 2, we discuss related work. In Section 3, we briefly describe Rabin cryptosystem and Arnold transform as these serve as the foundation of the proposed technique. In Section 4, we provide the details of the proposed technique for securing sensitive files using multi-image steganography. In Section 5, we present the results of experimentations and compare the proposed technique with some of the relevant techniques, and finally we conclude the paper by summarizing the research outcomes in Section 6.

2 Related work

To enhance the security of sensitive secret information, we resort to secret sharing, by way of segmenting the secret information and sharing among several members [46]. For (k,n) secret sharing, the secret message is revealed only when at least k out of n shares are present. (n,n) Secret sharing is a special case of the general secret sharing and it requires all the shares to unlock the secret.

Researchers have developed several mechanisms to achieve secret sharing. Shamir's interpolation method [46] was extended for sharing secret images, also called visual secret sharing (VSS) without any cryptographic computation [42]. In VSS, an image is broken into n shares/shadows, any k of the n users stack their shares to reveal the secret, but any $(k-1)$ or fewer users do not get any advantage. VSS can also be used for multiple images [17]. Although initially binary images were used for VSS, schemes for sharing the color images were developed subsequently [11, 21, 50].

Some techniques used for share generation are based on Chinese remainder theorem [23, 45]. Wu et al. [53] gave secret image reversible sharing scheme based on cellular automata. While some authors proposed that the shares generated were embedded in multiple images to give an appearance of a meaningful shadow so that no one gets suspicious about the secret message [34, 35], others came up with the concept of secret sharing with steganography and authentication to provide meaningful shares [33, 52]. Chang et al. [10] improved secret sharing using steganography by increasing capacity by hiding only $t-3$ secret bits, a significant improvement over proposals that involved hiding $t-1$ bits. Some works proposed invertible secret image sharing with steganography [34, 48, 52]. In these works, cover image was retrieved without distortion along with secret image. Tripathi et al.[49] proposed a sorted index code and a polynomial-based threshold secret sharing scheme to produce relevant shadows to achieve higher embedding capacity, yielding complete recovery of the cover image.

Chen and Wu [12] proposed a Boolean-based multi-secret image sharing scheme. It uses a random image generating function to generate random images from the secret images and boolean calculations are applied to share n secret images among n shared image. In Chen and Wu's scheme, partial secret information is revealed from $n-1$ or fewer shares, Yang et al. [57] proposed a more secure modified (n,n) multi secret image sharing scheme. The proposed technique that included a permutation function to enhance randomness of the

shared images. Using steganography, Wu and Yang provided authentication and reversible cover image recovery for compressed images [52]. Wu et al. [54] proposed a scheme that achieved authentication and secret image sharing. Their schemes deployed a polynomial-based technique that generated secret shares which were concealed in the color palettes of the cover image. Liao et al. [32] proposed an adaptive payload distribution based on texture features among multiple images to achieve enhanced security.

Kabirirad and Eslami [22] proposed a technique to generate pseudo-random shares using boolean operations to hide multiple images. Thus, it assured that no information about the secret image is revealed from the knowledge of $n - 1$ or fewer shares unlike in earlier boolean approaches. Logeshwari and Parvathy [36] proposed a secure (n, n) multi-secret sharing scheme based on the logistic chaotic sequence generated using a secret geometric pattern that was XORed with the image, thus providing enhanced security against unauthorized users. Gutub and Al-Ghamdi [18] proposed image based steganography to hide (n,n) secret shares produced using a counting-based technique. In a subsequent paper, Alkhodaidi and Gutub [6] proposed another counting-based secret sharing scheme which dynamically selects locations to embed secret shares based on share key size and cover image size. Using univariate and bivariate polynomials for secret sharing, Meng et al. [40] proposed a scheme, that thwarts the illegal participant attack. Secret could be recovered only if all the participants had valid shares. Using polynomial interpolation, Sardar and Adhikari [44] proposed a secret sharing approach for gray scale images. Further, to ensure that there is no leakage of secret information, the polynomial coefficient corresponding to the highest exponent was chosen randomly for introducing randomness within the shadow images. This also obviated the need for pre-processing to introduce randomness. Agarwal et al. [5] detected all the objects present in an image and applied (n,n) secret sharing scheme over the detected objects instead of the entire image, thus reducing the cost of secret sharing which is achieved using strict skewed binary tree.

3 Background

In the proposed approach for securing digital information, Rabin cryptosystem is applied for encrypting data and Arnold transform is used for randomizing the encrypted data. In this section, we briefly describe Rabin cryptosystem and Arnold transform.

3.1 Rabin cryptosystem modified

Unique prime factorization theorem says that any integer n, where $n > 1$ is either prime number or a product of prime factors. This factorization is unique in the sense that any two such factorizations differ only in the order in which the primes are multiplied. If the number has only small factors, then it is not hard to factorize. But it is well known that if the number is very large (for example 512 bits or longer) and has only a few large factors, then there is no efficient algorithm available to factorize. So, we have used the concept of factorization of positive integer in two-dimensional data for security. Rabin cryptosystem, is a variant of RSA wherein the value of exponential factor is fixed as two [43].

3.1.1 Encryption at sender's side

We first divide the secret data into chunks (M) based on the size of the public key (N). The size of one chunk is less than half of the key length. Pre-padding of the message is to be

done so that the message size remains consistent. Each chunk is concatenated with itself, and the encrypted message (C_t) is obtained using (1).

$$C_t = M^2 \pmod{N} \tag{1}$$

3.1.2 Decryption at receiver's end

While decrypting the cipher message, we pre-pad the cipher text based on the key length and then divide the cipher text into chunks. Next, we apply decryption formula to obtain four possible candidates, and select the one whose left side matches the right side when divided into two halves. Suppose, p and q are two large prime numbers whose product is the public key N. The primes p and q are distinct and satisfy the following equations [1]:

$$p \equiv 3 \pmod{4} \tag{2}$$

$$q \equiv 3 \pmod{4} \tag{3}$$

Now, integers a and b are calculated using extended Euclidean algorithm using the equation

$$a * p + b * q = 1 \tag{4}$$

Decryption is carried out as follows:

$$x = c^{(p+1)/4} \pmod{p} \tag{5}$$

$$y = c^{(q+1)/4} \pmod{q} \tag{6}$$

$$r_1 = (a * p * s + b * q * r) \pmod{N} \tag{7}$$

$$r_2 = (a * p * s - b * q * r) \pmod{N} \tag{8}$$

$$r_3 = -r_1 \pmod{N} \tag{9}$$

$$r_4 = -r_2 \pmod{N} \tag{10}$$

3.2 Arnold transform

Arnold transform works on the principle of shearing i.e. shifting rows or columns from one position to another so as to obtain the scrambled data. The number of iterations, applied to displace data positions each time act as the key required to be known at the receiver end. Initially, digital data in one dimensional form is taken from the user and the size of data is stored in the first 32 bits. Next, we apply padding (if required) to make the size of data a perfect square of an integer. Now we transform the data in the form of a square matrix and apply Arnold transform using (11), (12), (16).

$$\begin{bmatrix} xTransf \\ yTransf \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod l \tag{11}$$

$$\begin{bmatrix} xTransf \\ yTransf \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod l \tag{12}$$

At the time of decryption, inverse Rabin cryptosystem is applied to retrieve the binary vector, which is transformed into a square matrix and inverse Arnold transform is applied as per (13) and (17). We retrieve length of data from first thirty two bits and obtain the original data after discarding the extra bits.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} xTransf \\ yTransf \end{bmatrix} \mod l \tag{13}$$

[1] $a \pmod{n}$ is defined as set of all integers which have the same remainder as a when divided by n

3.3 Transforming coordinates

To implement Arnold transform, initially we create two matrices $xTrans_0$ and $yTrans_0$ and represent initial matrix indices, each of size $l \times l$.

$$xTrans_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & .. \\ 1 & 1 & 1 & 1 & .. \\ 2 & 2 & 2 & 2 & .. \\ 3 & 3 & 3 & 3 & .. \\ : & : & : & : & .. \end{bmatrix}_{l \times l} \tag{14}$$

$$yTrans_0 = \begin{bmatrix} 0 & 1 & 2 & 3 & .. \\ 0 & 1 & 2 & 3 & .. \\ 0 & 1 & 2 & 3 & .. \\ 0 & 1 & 2 & 3 & .. \\ : & : & : & : & .. \end{bmatrix}_{l \times l} \tag{15}$$

In each iteration x and y coordinate positions are transformed as:

$$\begin{aligned} (xTrans_{i+1}, yTrans_{i+1}) \rightarrow (((2.xTrans_i + yTrans_i) \mod l), \\ ((xTrans_i + yTrans_i) \mod l)) \end{aligned} \tag{16}$$

where $xTrans_{i+1}$ and $yTrans_{i+1}$ represent the values at x and y coordinate positions obtained after $(i+1)^{th}$ iterations of Arnold transformation respectively. $xTrans_i$ and $yTrans_i$ represent values at x and y coordinates obtained after i^{th} iteration of Arnold transformation respectively. Finally, the values are assigned to the final positions obtained after applying the specified Arnold transformations.

Similarly, for Inverse Arnold transform, with each iteration x and y coordinate positions are transformed as:

$$\begin{aligned} (x'Trans_{i+1}, y'Trans_{i+1}) \rightarrow (((x'Trans_i - y'Trans_i) \mod l), \\ ((-x'Trans_i + 2.y'Trans_i) \mod l)) \end{aligned} \tag{17}$$

where $x'Trans_{i+1}$ and $y'Trans_{i+1}$ represent values of x and y coordinate positions obtained after $(i+1)^{th}$ inverse iterations of Arnold transformation. $x'Trans_i$ and $y'Trans_i$ represent values of x and y coordinates obtained after i^{th} inverse iterations of Arnold transformation. So, final values are assigned to the positions after specific number of iteration key.

4 Proposed approach based on multi-image steganography

The input data is first encrypted using the public key of the receiver by employing the Rabin cryptosystem which works on the principle of asymmetric public key encryption mechanism.

The multi-image steganography enables hiding the data in the edges of multiple images. A threshold is chosen depending on the size of data to be hidden and the variation in pixel intensity values which determines the capacity of image. Each image hides an image header segment and payload. Each image header segment consists of its sequence number and a part of the randomized header information (partial header). Each partial header consists of a part of randomized header, where header consists of data length (32 bits) and part of message digest obtained from MD5(64 bits) and some random bits (salt) as shown in

Fig. 1. Data portions to be hidden in different images are created using minimal key and distribution sequence key. Minimal key is the minimum data to be hidden in every image and thus, ensures some portion of the secret data is distributed in each stego image. Distribution sequence key is a sequence of numbers which represents the maximum contiguous data that can be hidden in each image at a time. Data portions are hidden in each image according to the distribution key in round robin fashion.

Figure 2 demonstrates an example of multi-image steganography. In this example, we have taken 3 cover images. Header segment for each image is 40 bit long and includes sequence number and partial header. Minimal key is taken as 3 to ensure that minimum 3 bits of data are hidden in each image. Distribution sequence key is taken as sequence of: 10, 15, 20, 5. To retrieve and authenticate data in multi-image steganography scheme, partial header is extracted from each image and arranged according to the obtained sequence numbers to obtain randomized header segment. It is rearranged, according to the PRNG used in the hiding phase, to get the header segment. Data length and message digest are extracted from this header segment. Data is extracted from the edges of the images and rearranged using minimal key and distribution sequence key as mentioned above. Message Digest is calculated from the retrieved data and authenticated with the message digest retrieved from the header sequence. In the proposed technique, all shares are required to extract the data. This technique also detects any share that is corrupted.

4.1 Sequence of operations to encrypt and hide in multiple images

Initially, we read the the secret file that may be in text, image, audio or video form. Public key encryption mechanism — Rabin cryptosystem is used to encrypt it using public key (N) of the receiver. Encrypted data is padded suitably and randomized using Arnold transform.

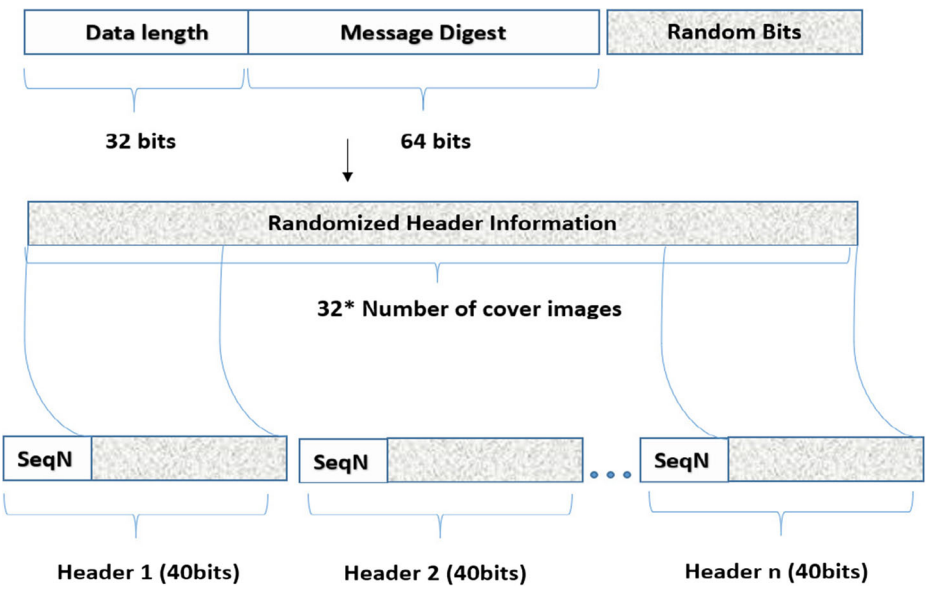

Fig. 1 Creating header information for each image

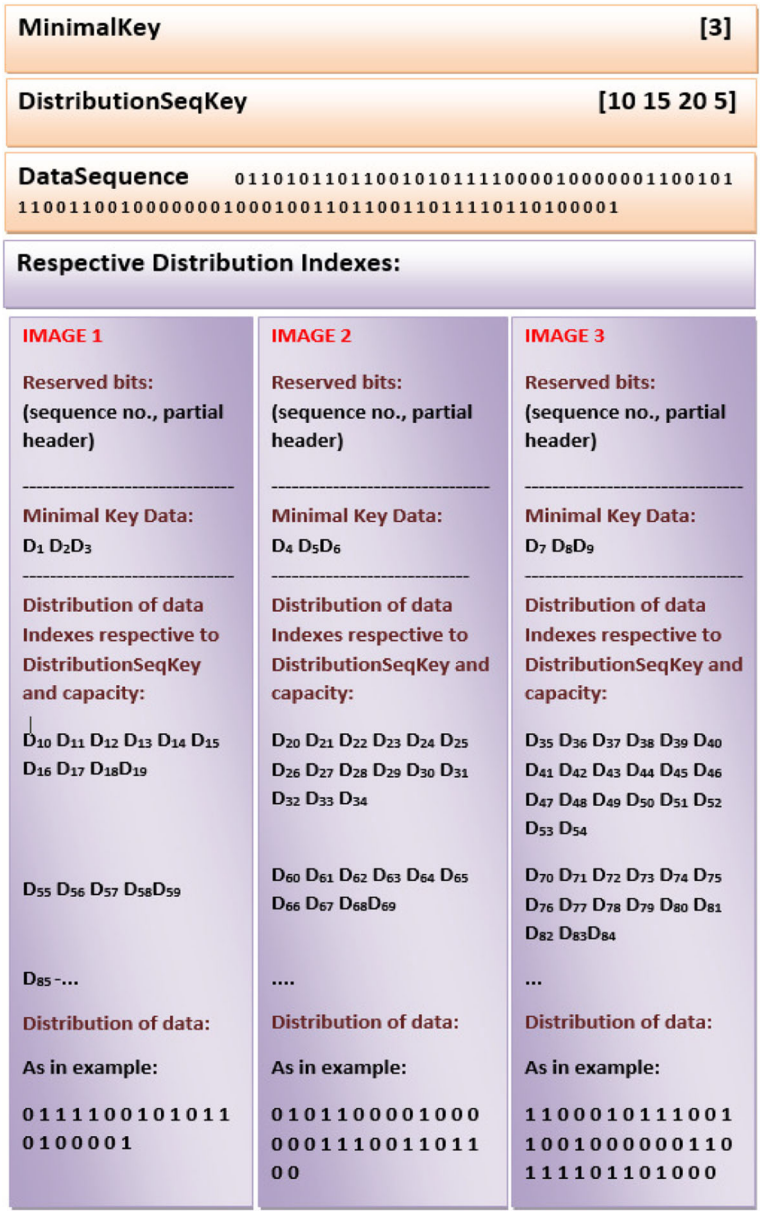

Fig. 2 An example to demonstrate multi-image steganography using 3 cover images

Arnold transform shifts rows and columns to displace data iteratively for specified number of iterations. Length and message digest of this randomized encrypted data are concatenated and further randomized to create header information. Thus, header for each image is composed of its sequence number and part of this header information, as shown in Fig. 1. Header information for each image along with randomized encrypted data are hidden, first using the minimal key and then using the distribution key in the edges of the multiple

images using adaptive edge technique using the LSBMR technique [37]. Thus, information is shared among several images. Procedure to encrypt and hide the secret data shares in multiple images is given in Fig. 3. Algorithmic description for multi-image steganography and authentication is given in Algorithms:1 and 4.

Algorithm 1 Algorithm for multi-image steganography and authentication.

1: ▷ **Input: Cover images**$(C(1), C(2), ..., C(n))$**, data(d), RabinPublicKey(N), ArnoldKey(iter), headerSeed(headrK), minimalKey(minK), distributionKey(distrK)**
2: ▷ Output: Crypto-stego images$(CSI(1), CSI(2), ..., CSI(n), t)$
3: **procedure** **CreateCryptoStegoImg**$(C(1), C(2), ..., C(n), d, N, iter, headrK, minK, distrK)$
4: $(d_r) \leftarrow$ Apply Rabin(d, N)
5: $(d_{ar}) \leftarrow$ Apply Arnold$(d_r, iter)$
6: $[CSI(1), CSI(2), ..., CSI(n), t] \leftarrow$ creatingDataSharesAndHiding$(d_{ar}, headrK, minK, distrK)$
7: Return Crypto-stego images$(CSI(1), CSI(2), ..., CSI(n), t)$

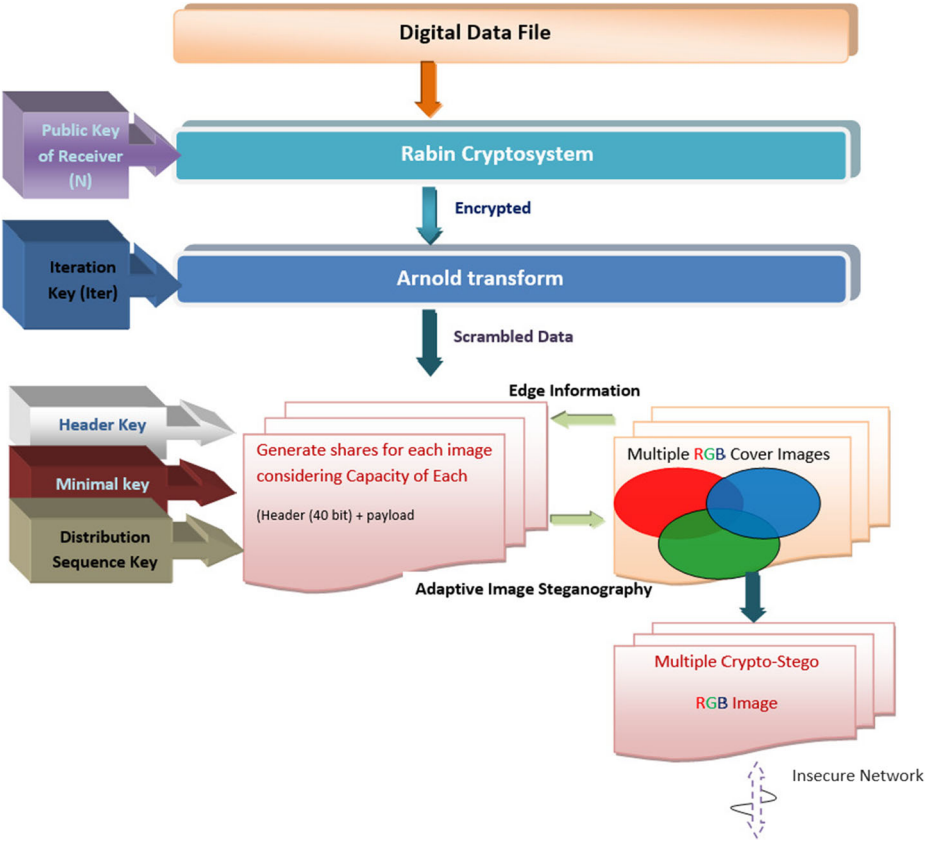

Fig. 3 Procedure to encrypt and hide the original data in multiple images

Algorithm 2 Function: Creating data shares and hiding.

▷ **Input:Cover images** $C(1), C(2), ..., C(n)$**, shuffled encrypted data(d), header Seed(headrK), minimal Key(minK), distribution Key(distrK)**
▷ Output: Crypto-stego image $CSI(1), CSI(2), ..., CSI(n), threshold(t)$
procedure **CreatingDataSharesAndHiding**$(C(1), C(2), ..., C(n), d, N, iter, headrK, minK, distrK)$
    $md \leftarrow$ message digest(d)
    $(lm) \leftarrow$ len(d)
    $(lr) = (lm \parallel md)$
    ▷ create header for each image
if $(NumOfCIs <= 3)$
    $headerLen = len(lr)/NumOfCIs$
else
    Append random bits to lr, such that $len(lr) = 32 * n$
    $headerLen = 32$
    $(lr) = (randomize(lr, headrK))$
for $i = 1 : NumOfCIs$
    $in = 0$
    $header(i) = sequenceNo(i) \parallel lr[in : headerLen]$
    $in = in + 32$
    ▷ create data shares for edges
    $Cap(i) =$ Capacity of $CI(i)$ at threshold t
while $(len(md) + len(lr)) > \sum Cap(i)$
    Recalibrate threshold t and recalculate capacity C(i)
    $ds(1), ds(2), ..., ds(n) =$ Distribute d using MinKey and distrK Section:4
for $i = 1 : n$ **do**
    $in = 1$
    $s(i) = header(i) \parallel ds(i)$
    $CSI(i) =$ Hide data in $CI(i)$ edges using adaptive edge LSBMR$(S(i))$
    Return Crypto-stego images $CSI(1), CSI(2), ..., CSI(n)$ and calibrated threshold t

4.2 Sequence of operations to retrieve decrypted data from multiple images

At the receiver end, user retrieves payload from the edges of each image and then arranges them according to the original data sequence in correspondence to distribution sequence indexes obtained by minimal data key and distribution sequence key. Message digest of the data obtained is calculated and matched with decrypted message digest to detect any manipulation of data, thus keeping a check on authenticity of data. Inverse Arnold transform is applied to rearrange the data properly and then inverse Rabin cryptosystem using private key of receiver as in asymmetric public key system is applied to decrypt the data in original form. Procedure to extract data from multiple images and decrypt the secret data is given in Fig. 4. Algorithmic description for decryption is given in Algorithms 3 and 4.

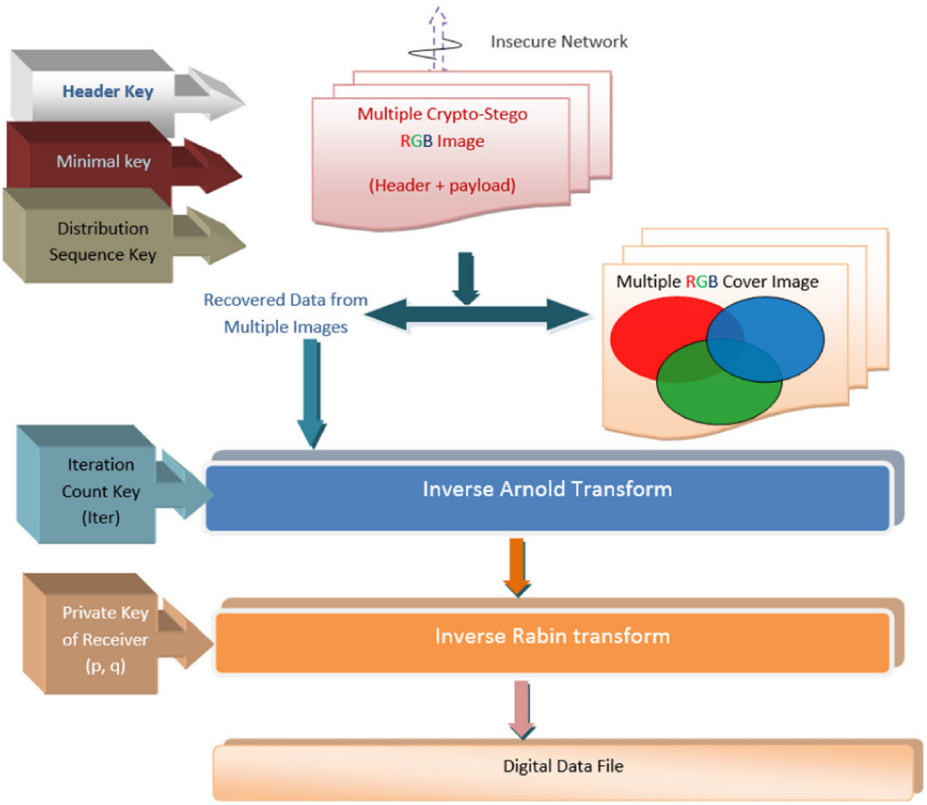

Fig. 4 Procedure to extract data from multiple images and decrypt to obtain the original Information

Algorithm 3 Algorithm for extraction and decryption in multi-image steganography and authentication.

▷ **Input: Crypto-stego images**($CSI(1), CSI(2), ..., CSI(n)$)**, RabinPrivateKeys(p,q), ArnoldKey(iter), headerSeed(headrK), minimalKey(minK), distributionKey(distrK), calibrated threshold(t)**

▷ Output: Secret data(d)

procedure ExtractData($CSI(1), CSI(2), ..., CSI(n), p, q, iter, headrK, minK, distrK, t$)

    $d_{ar} \leftarrow$ ExtractDataFromShares(CSI(1),CSI(2),...,CSI(n),headrK,minK,distrK,t)

    $d_r \leftarrow$ Apply Inverse Arnold($d_{ar}, iter$)

    $d \leftarrow$ Apply Inverse Rabin(d_{ar}, p, q)

    Return Authenticated secret d

Algorithm 4 Function: Extract data from shares.

▷ Input:Cover images $CSI(1), CSI(2), ..., CSI(n)$, headerSeed(headrK), minimalKey(minK), distributionKey(distrK), calibrated threshold(t)
▷ *Output: Secret data(d)*
procedure ExtractDataFromShares$(CSI(1), CSI(2), ..., CSI(n), headrK, minK, distrK, t)$
    if $(NumOfCIs <= 3)$ **then**
        $RHeaderLen = [96)/NumOfCIs] + 8$
    else
        $RHeaderLen = 40$
    for i = 1:NumOfCRSs **do**
        $rHeader(i) \leftarrow ExtractrHeaderLenamountofdatafromCSI(i)$
        SequenceNo(i) = rHeader(i)[0:7]
        partialHeader(i) = rHeader(i)[8:headerLen]
    lr = Merge partialHeader(i) in order of Sequence no(i) of images in sorted form
    header = Reorder(lr,headrK)
    dataLen = header[0:31]
    mdHidden = $header[32 : 96]$
    dataSize(i) = Calculate data size using minK, distrK using Section:4
    data(i) = Extract data of size dataSize(i) from CSI(i) using adaptive edge LSBMR technique
    mdD = Calculate message digest(d)
    Authenticate mdD using $mdHidden$
    Return hidden data (d)

5 Experimental results

To evaluate the proposed authentication scheme, we carried out extensive experimentation using audio, video, and image data. Based on histogram analysis, pixel intensity correlation graphs, and MSE, PSNR, and correlation coefficient values, we demonstrate that presented technique is suitable for safe transmission of digital data.

5.1 Embedding an audio clip

In this section, we have shown the results of experiments for a sample audio file. We have used a small audio clip of size $10.6Kb$ (Fig. 5a) as the secret information and Fig. 5b, c, and d as the cover images. On hiding the encrypted audio file in the cover images, we obtain the crypto-stego images shown in Fig. 6a, b, c respectively. The recovered audio file showed no perceptible distortion.

Histogram Analysis Histogram of the cover images in Fig. 5b, c, and s (see Fig. 7a, b, c respectively) resemble histogram of crypto-stego images Fig. 6b, c, and d (see Fig. 8a, b, and c). It is evident that the pair of the histograms of images before and after embedding the secret information show so much similarity that one can not guess that any data is hidden in cover images.

Pixel intensity Correlation Graphs Pixel intensity correlation graphs of cover images match the corresponding pixel intensity correlation graphs of crypto-stego images. For example, pixel intensity correlation graphs of cover image 1 in horizontal, vertical and diagonal direction shown in Fig. 9a, b, and c respectively are quite similar to the pixel intensity correlation graphs of crypto-stego image 1 in horizontal, vertical and diagonal direction shown in Fig. 11a, b, and c respectively. Similar trend was seen for the correlation graphs

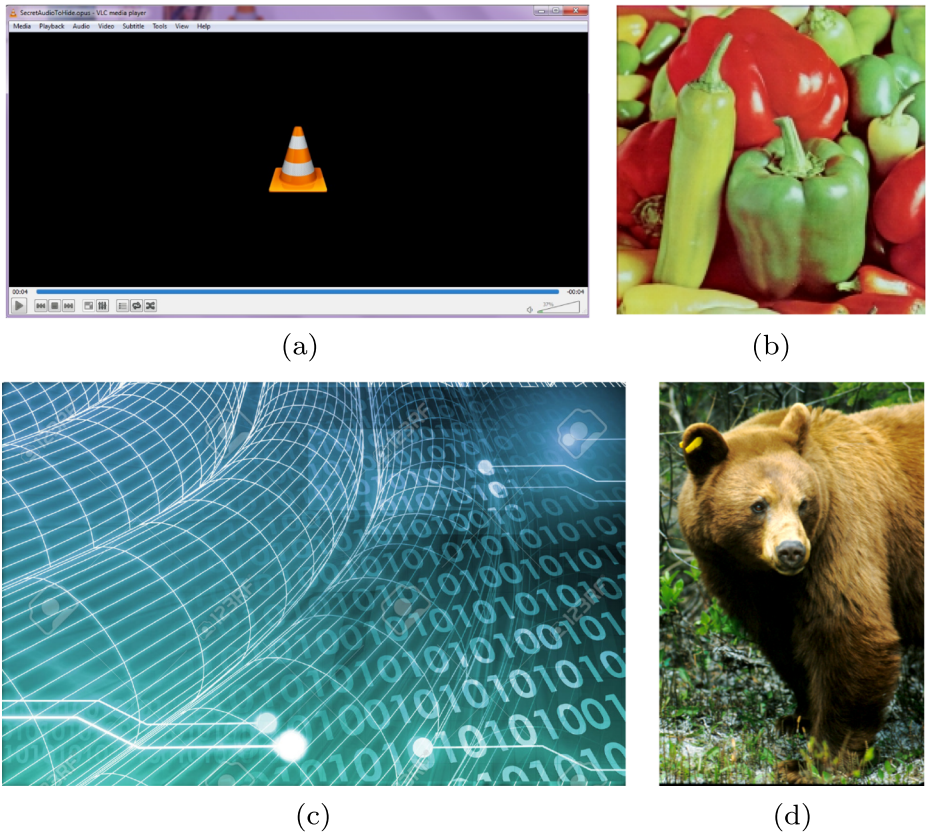

Fig. 5 Data used in experiments: **a** Original audio data **b** Cover image 1 **c** Cover image 2 **d** Cover image 3

of cover images 2 and cover image 3 (Figs. 9 and 10), and their respective crypto-stego images (Figs. 11 and 12).

MSE, PSNR, Correlation Coefficient In Table 1, we present the correlation coefficient, MSE, and PSNR values between the pairs or original cover images and the corresponding crypto-stego images. Table shows that the correlation coefficient values are 1 (up to four places of decimal), MSE values are low, and PSNR values are high for all pairs of original cover images and the corresponding crypto-stego images, implying that detection of the presence of any significant information is unlikely.

5.2 Embedding a video clip

In this section, we have shown the results of experiments for a sample video file. We have used a small video clip of size $285Kb$ and 00:00:12 (Fig. 13a) as the secret information and Fig. 13b, c, and d as the cover images. On hiding the encrypted audio file in the cover images, we obtain the crypto-stego images which resemble original cover images. The recovered video file showed no perceptible distortion (Fig. 14).

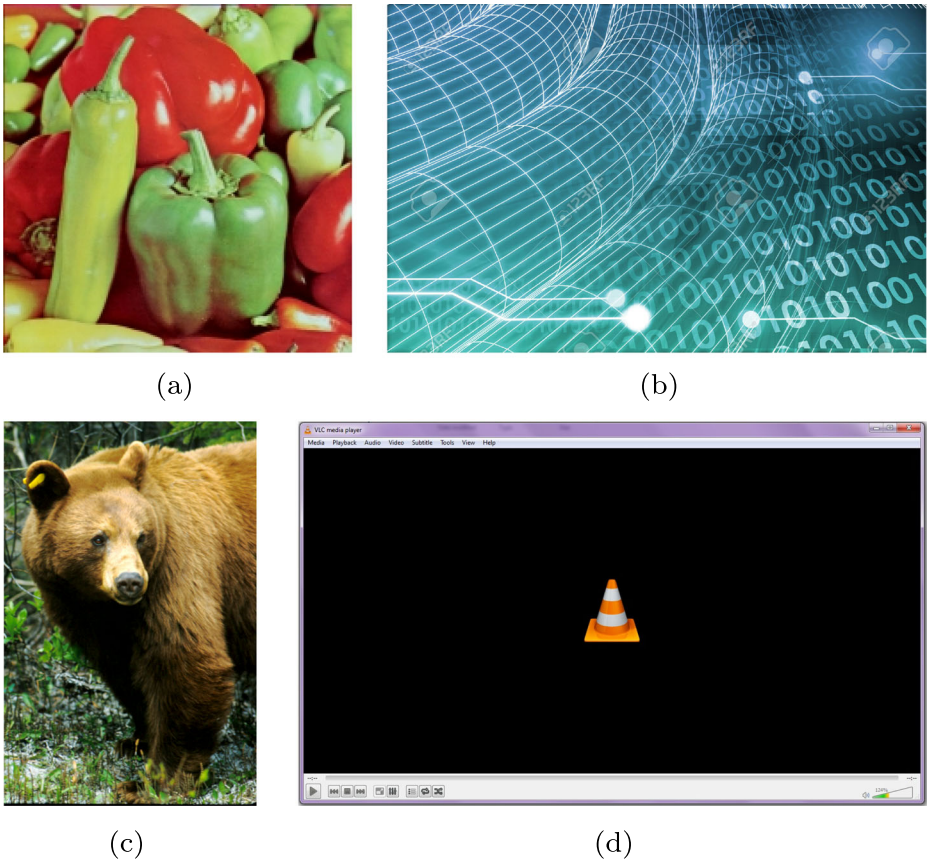

Fig. 6 Experimental results of audio data: **a** Crypto-stego image 1 **b** Crypto-stego image 2 **c** Crypto-stego image 3 **d** Recoverd audio

Histogram Analysis Histogram of the cover images in Fig. 13b, c, and d (see Fig. 15a, b, c respectively) resemble histogram of crypto-stego images Fig. 16a, b, and c. It is evident that the pair of the histograms of images before and after embedding the secret information show so much similarity that one can not guess that any data is hidden in cover images. Similar to audio results, for video data also pixel intensity correlation graphs obtained for cover image

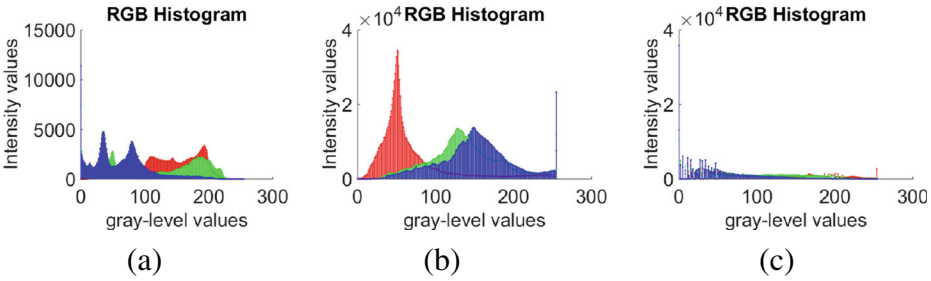

Fig. 7 Histograms of cover images: **a** Cover image 1 **b** Cover image 2 **c** Cover image 3

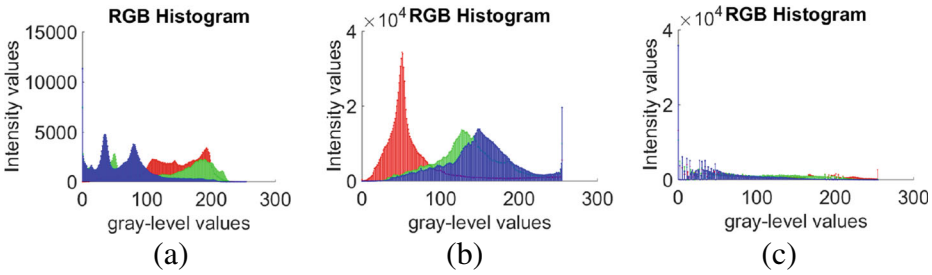

Fig. 8 Histograms of crypto-stego images: **a** Crypto-stego image 1 **b** Crypto-stego image 2 **c** Crypto-stego image 3

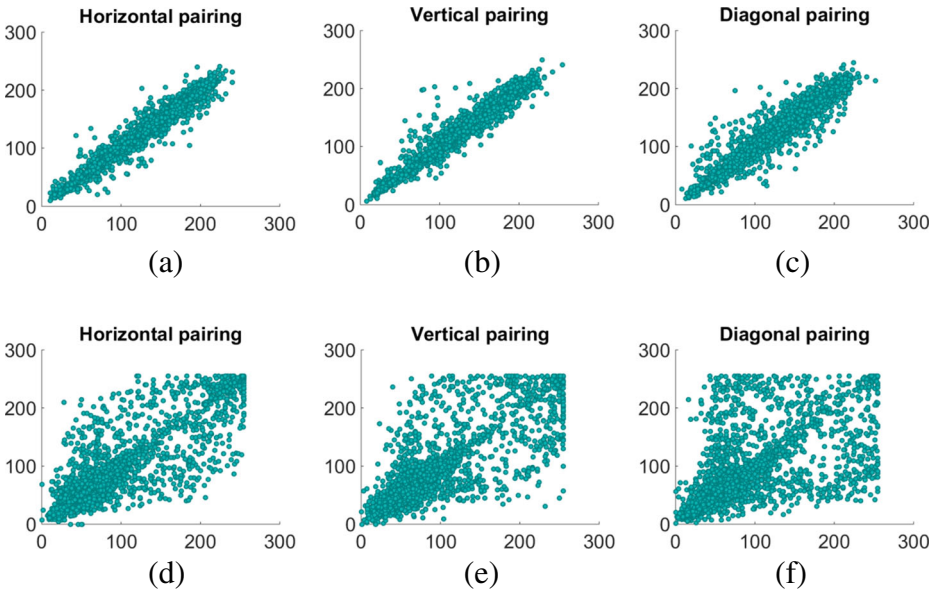

Fig. 9 Pixel intensity distribution of cover images of audio data: **a** At Horizontal direction of cover image 1 **b** At vertical direction of cover image 1 **c** At Diagonal direction of cover image 1 **d** At Horizontal direction of Cover image 2 **e** At Vertical direction of Cover image 2 **f** At Diagonal direction of Cover image 2

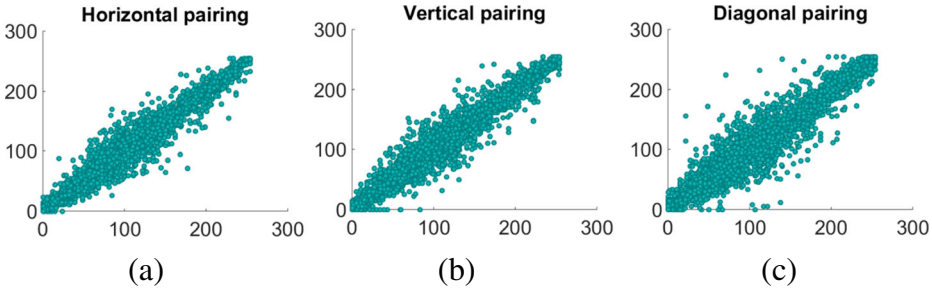

Fig. 10 Pixel Intensity Distribution of cover image of Audio data: **a** At Horizontal direction of cover image 3 **b** At Vertical direction cover image 3 **c** At Diagonal direction of cover image 3

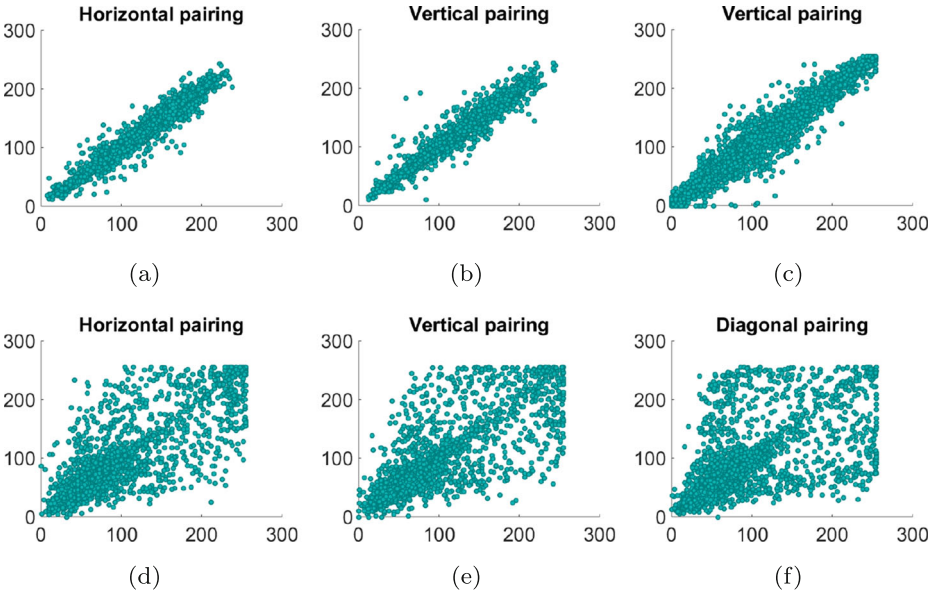

Fig. 11 Pixel Intensity Distribution of crypto-stego image of Audio data: **a** At Horizontal direction of crypto-stego image 1 **b** At Vertical direction of crypto-stego image 1 **c** At Diagonal direction of crypto-stego image 1 **d** At Horizontal direction of crypto-stego image 2 **e** At Vertical direction of crypto-stego image 2 **f** At Diagonal direction of crypto-stego image 2

Table 1 Statistical analysis for Audio data at edge threshold 26

S.No.	Input images	Correlation	MSE	PSNR
1.	Between Cover image 1(Fig. 5b) and Crypto-stego image 1(Fig. 6a)	1.0000	0.0088	68.6707
2.	Between Cover image 2 (Fig. 5c) and Crypto-stego image 2(Fig. 6b)	1.0000	0.0582	60.4850
3.	Between Cover image 3 (Fig. 5d) and Crypto-stego image 3(Fig. 6c)	1.0000	0.0299	63.3725

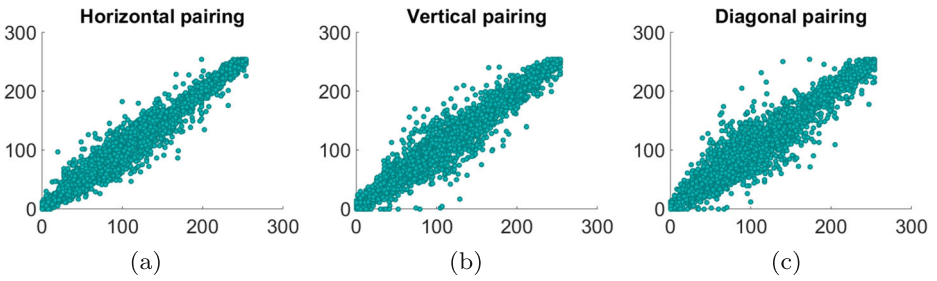

Fig. 12 Pixel Intensity Distribution of crypto-stego image of Audio data: **a** At Horizontal direction of crypto-stego image 3 **b** At Vertical direction of crypto-stego image 3 **c** At Diagonal direction of crypto-stego image 3

Fig. 13 Data used in experiments: **a** Original video data **b** Cover image 1 **c** Cover image 2 **d** Cover image 3

in horizontal, vertical and diagonal direction resemble the pixel intensity correlation graphs in horizontal, vertical and diagonal direction respectively of crypto-stego image. This shows robustness to any kind of attack as the adversary is not aware of presence of any external data.

MSE, PSNR, Correlation Coefficient In Table 2, we present the correlation coefficient, MSE, and PSNR values between the pairs of original cover images and the corresponding crypto-stego images. The table shows that the correlation cofficient values are 1 (up to four places of decimal), MSE values are low, and PSNR values are high for all pairs of original cover images and the corresponding crypto-stego images, implying that detection of the presence of any significant information is unlikely.

5.3 Embedding images

In this section, we have shown the results of hiding hidden images in multiple cover images. Images used as cover images and secret data are taken from SIPI database: http://sipi.usc.edu/database/database.php?volume=misc and Peticolas database: https://www.petitcolas.net/watermarking/image_database/. Figure 17a, b, and c show average PSNR, MSE and SSIM graphs respectively on varying number of cover images in the range 3 to 15 for hiding five color images from dataset bear ($394 \times 600(9,00,111\ bits)$), opera ($695 \times 586\ (16,81,429\ bits)$), papermachine ($200 \times 132(1,08,826\ bits)$), wildflowers ($200 \times$

Fig. 14 Experimental results of Video data: **a** Crypto-stego image 1 **b** Crypto-stego image 2 **c** Crypto-stego image 3 **d** Recoverd video

135(1, 14, 318 $bits$)) and baboon (512 × 512(11, 34, 554 $bits$)). As the number of cover images increases, the size of the data to be hidden in edges also increases as more space is available. The cover images resemble camouflage images produced and imperceptibility is higher as the number of cover images is more. In Fig. 17d, a set comprising three randomly chosen images is used as the cover to hide an image (Pills of size 200 × 130 from the Peticolas database). The experiment is repeated ten times with different sets of cover images and the PSNR value is calculated each time. Images are selected from peticolas dataset with different features like bright color, high texture, fine details, lines and edges. Although the boxplot of PSNR values (Fig. 17d) shows some variation for different choices of cover images, nevertheless PSNR values remain high (> 53).

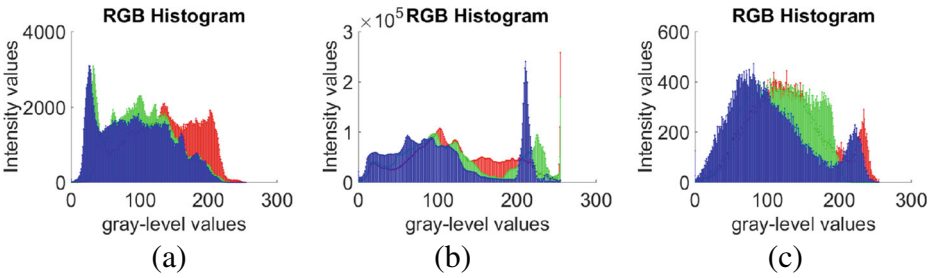

Fig. 15 Histogram for cover image of Video data : **a** Cover image 1 **b** Cover image 2 **c** Cover image 3

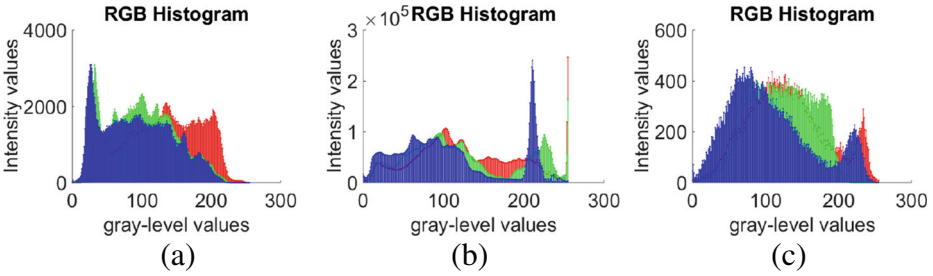

Fig. 16 Histogram for crypto-stego image of Video data : **a** Crypto-stego image 1 **b** Crypto-stego image 2 **c** Crypto-stego image 3

Table 2 Statistical details relating to Video data at threshold 4

S.No.	Input images	Correlation	MSE	PSNR
1.	Image with Seq. No. 1(Fig. 13b and Fig. 14a)	1.0000	0.2026	55.064
2.	Image with Seq. No. 2 (Fig. 13c and Fig. 14b)	1.0000	0.1878	55.391
3.	Image with Seq. No. 3 (Fig. 13d and Fig. 14c)	0.9999	0.3078	53.2485

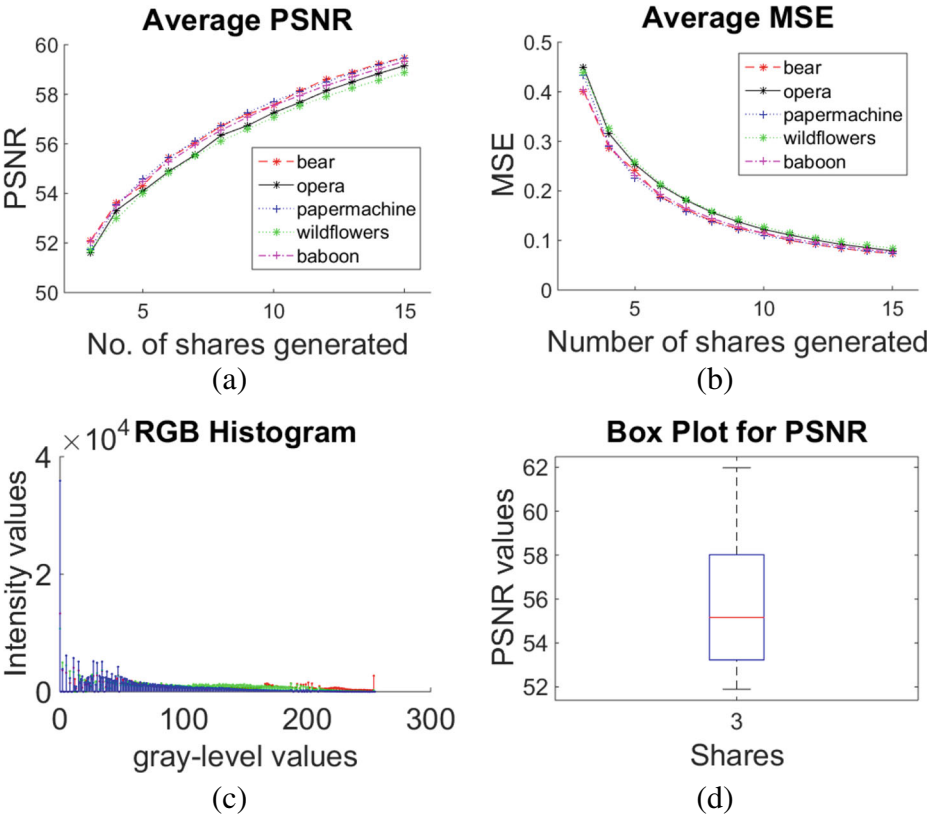

Fig. 17 Data characteristics set **a** Average PSNR graph **b** Average MSE graph **c** Average SSIM graph **d** Boxplot for different cover images

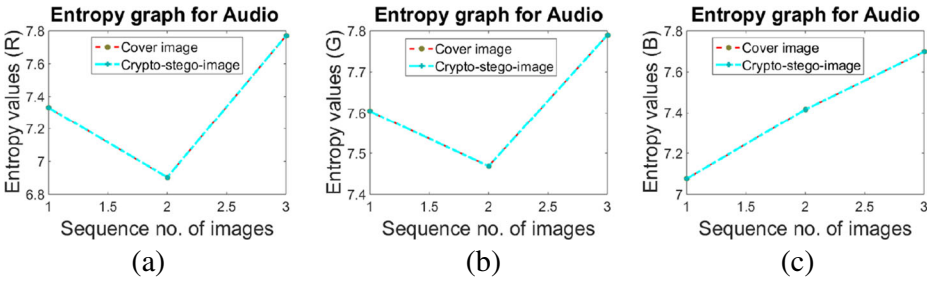

Fig. 18 Entropy Graph of Audio data: **a** Red Component **b** Green Component **c** Blue Component

5.4 Security issues

The proposed technique is CPA and CCA secure as initially asymmetric Rabin cryptosystem and then Arnold transform are applied, no extra information is required for correct decryption. Thus, the proposed technique does not reveal any information about roots that may lead to recovery of the prime numbers used. Further, Rabin cryptosystem key is as hard to break as factoring of large positive integer. The encrypted data is hidden behind random RGB color of a color image to veil the presence of any important data.

5.4.1 Key sensitivity

Key sensitivity is an important attribute to determine the strength of any security system. High key sensitivity is needed for secure cryptosystems, i.e. one should not be able to recover the original data without the exact keys. In the proposed technique, if any discrepancy is found at decryption stage, then the data cannot be recovered. If the used key differs from the original key even sightly, then the data cannot be recovered from crypto-stego image.

5.4.2 Entropy analysis

For a true random system that generates 256 symbols with an equal probability, the entropy is equal to 8. In image security, entropy for output image is closed to the entropy of the input image then information of output image is similar to the original data. If the entropy of encrypted data is smaller than the ideal value 8, the proposed cryptosystem is free from attack. Figure 18a, b, c are entropy graph of audio data for red, green and blue components

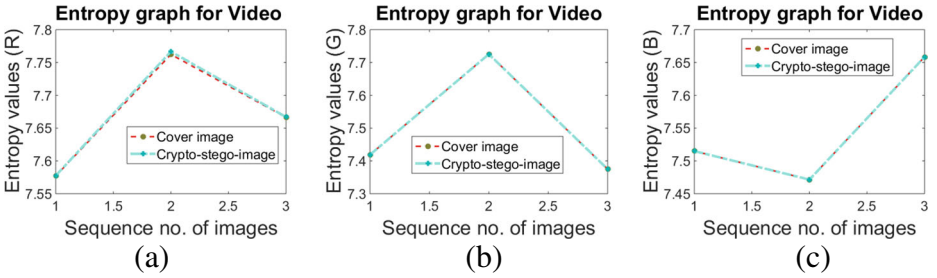

Fig. 19 Entropy Graph of Video data : **a** Red Component **b** Green Component **c** Blue Component

Table 3 Entropy values for Hidden Audio Data in cover image and crypto-stego image

S.No.	Sequence No.	R/G/B Component	Original cover image	crypto-stego image
1.	1	Red Component	7.33093	7.33119
2.	1	Green Component	7.60356	7.60356
3.	1	Blue Component	7.07543	7.07543
4.	2	Red Component	6.90306	6.90370
5.	2	Green Component	7.46868	7.46868
6.	2	Blue Component	7.41621	7.41621
7.	3	Red Component	7.77053	7.77088
8.	3	Green Component	7.78999	7.78999
9.	3	Blue Component	7.69841	7.69841

respectively for each of the cover image and crypto-stego image. Figure 19a, b, c are entropy graph of video data for red, green and blue components respectively, for each of the cover image and crypto-stego image. In both audio and video entropy graphs, the values obtained for cover image are shown in red color and crypto-stego image are shown in light blue color. As the two graphs nearly coincide with each other, the presence of information in stego-images will go unnoticed. In Tables 3 and 4, we summarize the entropy values for hidden audio data and video data in different RGB Components of each cover image and crypto-stego image. As the entropy values obtained for cover image red component are quite close to that of crypto-stego image, it is very difficult to judge any change in cover image. So, the proposed cryptosystem is appropriate for audio and video data.

5.5 Comparison with competing techniques

- Proposed technique provides functionality for applications such as shared passwords, where all the members are required to open the locked password. Secret can be digital information in form of text, image, audio or video. However, several state-of-the-art schemes [10, 11, 56, 57] are applicable for images only.
- The proposed technique is applicable as(n, n)secret sharing technique. Secret information is shared among all the shareholders, any one member cannot reveal any partial information until he/she gets access to all the multiple images and decryption, and

Table 4 Entropy values for Hidden Video Data in cover image and crypto-stego image

S.No.	Sequence No.	R/G/B Component	Original cover image	crypto-stego image
1.	1	Red Component	7.5770	7.5775
2.	1	Green Component	7.4189	7.4189
3.	1	Blue Component	7.5149	7.5149
4.	2	Red Component	7.7622	7.7666
5.	2	Green Component	7.7253	7.7253
6.	2	Blue Component	7.4715	7.4715
7.	3	Red Component	7.6663	7.6669
8.	3	Green Component	7.3748	7.3748
9.	3	Blue Component	7.6584	7.6584

Table 5 Comparison of PSNR Values

Parameters	Tripathi et al	Proposed approach
Tank image Fig. 20a	34.12	53.8
Truck image Fig. 20b	34.82	53.64
Zelda image Fig. 20c	35.47	53.0247

hiding keys and algorithm to retrieve them. In [12] multi-image secret sharing scheme, partial secret information can be revealed from $n-1$ or fewer shared images [57].

- Proposed technique provides authentication of data. Even if there is any manipulation of data over insecure network, message digest of received data would not match the message digest created at sending time. Techniques [48, 55, 57] do not provide authentication of data.

- Most of the earlier developed techniques provide steganography [13, 48, 56] for image data. The schemes [4, 4, 14] provide security through cryptographic approach only. The proposed crypto-stego technique provides multi-layer security to large data in multiple images based on cryptographic and steganographic approach.

- The proposed technique uses Rabin cryptosystem for encryption which is considered as hard to break as factoring a number. Further bit operations for encryption in cryptographical part is $O((logN)^2)$, where N is public key for encryption and $logN$ represents minimum length of its bitwise representation. In this approach, the cryptographical part is fast as the algorithm used is simple and has low complexity.
- It is chosen ciphertext attack secure as it does not require delivery of any information at receiver's side. Also, there is no need to pass any known string of bits in data itself as it makes data prone to give some information regarding key on attack. Further, no need for any semantic analysis to identify the correct root. But earlier techniques [9, 20, 39], required delivery of information at receivers side or semantic analysis of root to retrieve the correct root. Thus, these techniques were vulnerable to chosen plaintext attack and chosen ciphertext attack.
- Proposed technique produces high quality image as data is hidden edges of cover images adaptively. Crypto-stego images produced are highly imperceptible.

The Tables 5, 6 and 7 show the comparison of the proposed approach with [49] and [6] respective to various images in Fig. 20 on quantitative measures like PSNR, SSIM, effect of different data sizes on PSNR. Results are obtained on average values of shadow images produced for (3,3) secret sharing. It is observed that proposed approach outperforms other state of the art techniques.

Table 8 shows comparison of the proposed techniques with its competitors on various parameters. Researchers have used different approaches to achieve secret sharing as shown

Table 6 Comparison of SSIM Values

Image	Tripathi et al.	Proposed approach
Tank image Fig. 20a	0.8819	0.9984
Truck image Fig. 20b	0.9170	0.9985
Zelda image Fig. 20c	0.9274	0.9973

Table 7 Effect of data size on a sample image (Monarch: Fig. 20d) on PSNR values

Data Size	AlKhodaidi and Gutub	Proposed approach
64bits	87.09	88.65
256bits	83	86.18
1024bits	76.12	82.885
3072bits	75	79.31

in the table. It shows how the respective schemes work for different parameters. The column *Technique used* document the techniques used for secret sharing (SS). The column *Meaningful shares* depicts whether the secret shares are represented as meaningful images. The column *Authentication* depicts whether a scheme supports authentication. The column *PSNR* depicts imperceptibility level. The column *Secure* provides information about security level (leakage if any, immunity to attacks like RS steganalysis) and applicability. Finally, the column *sensitivity* indicates level of robustness to identify minor modifications. It may be observed that the proposed technique passes all the requirements of meaningful shares, authentication, high PSNR, applicability to various forms of data like audio, video, and images, infeasibility of recovery from partial shares, resistant to RS detection and high sensitivity.

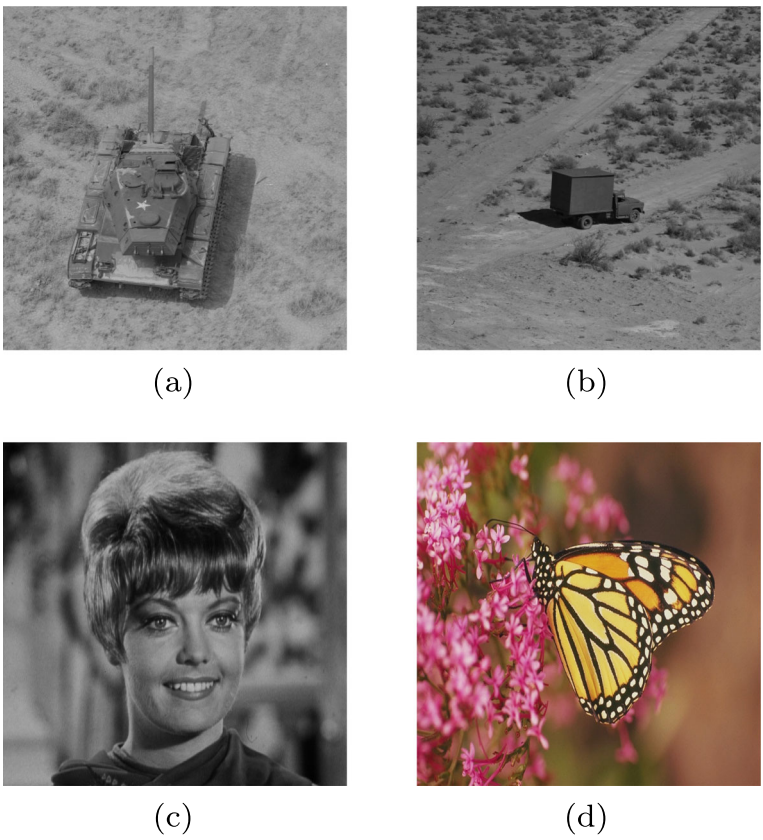

Fig. 20 Figures used for comparison **a** Tank **b** Truck **c** Zelda **d** Monarch

Table 8 Comparison of the proposed crypto-stego system with relevant papers

Schemes	Technique Used	Meaningful shares	Authentication	PSNR (in dB)	Secure	Sensitivity
Tang et al. [48]	high capacity stegano	True	False	36.98	-	-
Sun [47]	Edge based stegano	True	-	63.48	-	-
Chen and Wu [12]	Boolean based SS	False	False	-	Insecure	Low
Yang et al. [57]	Boolean based SS	False	False	-	Prevents partial information retrieval from fewer shared images	high
Liu et al. [35]	Progressive SS	True	-	24.79	Resistant to RS detection	High
Wu and Yang [52]	Polynomial based SS	True	True	24.66	-	-
Kabirirad [22]	Boolean based SS	-	-	-	Prevents partial information retrieval from fewer shared images	High
Gutub [18]	Counting based SS	True	-	-	Secure	High
Logeshwari [36]	Logistic chaotic sequence based SS	False	-	-	Prevents partial information retrieval from fewer shared images	High
Sardar [44]	Polynomial based SS	False	-	-	Secure for gray scale images	High
Proposed technique	Crypto-stego technique	True	True	51.82	Applicable to text, image, audio or video, prevents partial secret retrieval from fewer shared images, Resistant to RS detection	High

6 Conclusion

The proposed (n, n) secret sharing scheme secures secret information in the form of text, image, audio, or video as shares hidden in multiple images. The secret information cannot be revealed even if the shares are compromised. Sensitive information is encrypted using Rabin cryptosystem and scrambled. Cipher message is hidden in multiple images using minimal and distribution sequence keys. The proposed technique ensures uniform distribution of secret information across multiple shares. Using the evaluation metrics PSNR, MSE, SSIM, and entropy, we have demonstrated a high level of imperceptibility achieved using the proposed technique. It has been successfully demonstrated that the proposed technique can withstand known plaintext and chosen ciphertext attacks. Sensitivity analysis reveals that even a minor variation in a single share makes the recovery of the secret message infeasible. Comparison with the state of the art techniques indicates that the proposed technique either scores over its competitors or performs equally well in terms of the standard evaluation metrics.

Acknowledgements Authors thank the referees for their comments and suggestions which improved the presentation of the paper. Prof. R. K. Sharma is ConsenSys Blockchain Professor. He thanks ConsenSys AG for that previlage. This work has been supported for University of Delhi, Delhi-110007 with research grant RC/2015/9677.

References

1. Abdulla AA (2015) Exploiting similarities between secret and cover images for improved embedding efficiency and security in digital steganography. PhD thesis, University of Buckingham
2. Abdulla AA, Sellahewa H, Jassim SA (2014) Steganography based on pixel intensity value decomposition. In: Mobile multimedia/image processing, security, and applications 2014, International society for optics and photonics, vol 9120, p 912005
3. Abdulla AA, Sellahewa H, Jassim SA (2019) Improving embedding efficiency for digital steganography by exploiting similarities between secret and cover images. Multimed Tools Appl 78(13):17,799–17,823
4. Abuturab MR (2012) Securing color information using arnold transform in gyrator transform domain. Opt Lasers Eng 50(5):772–779
5. Agarwal A, Deshmukh M, Singh M (2020) Object detection framework to generate secret shares. Multimedia Tools and Applications 1–22
6. AlKhodaidi T, Gutub A (2020) Refining image steganography distribution for higher security multimedia counting-based secret-sharing. Multimedia Tools and Applications 1–31
7. Bai J, Chang CC, Nguyen TS, Zhu C, Liu Y (2017) A high payload steganographic algorithm based on edge detection. Displays 46:42–51
8. Balkrishan J, Singh AP (2016) Concealing data in a digital image with multilayer security. Multimed Tools Appl 75(12):7045–7063
9. Buchmann J (1999) Introduction to cryptography. Springer, New York
10. Chang CC, Hsieh YP, Lin CH (2008a) Sharing secrets in stego images with authentication. Pattern Recogn 41(10):3130–3137
11. Chang CC, Lin CC, Lin CH, Chen YH (2008b) A novel secret image sharing scheme in color images using small shadow images. Inf Sci 178(11):2433–2447
12. Chen CC, Wu WJ (2014) A secure boolean-based multi-secret image sharing scheme. J Syst Softw 92:107–114
13. Chen H, Du X, Liu Z, Yang C (2015) Optical color image hiding scheme by using gerchberg–saxton algorithm in fractional fourier domain. Opt Lasers Eng 66:144–151
14. Chen L, Zhao D (2009) Color image encoding in dual fractional fourier-wavelet domain with random phases. Opt Commun 282(17):3433–3438
15. Chien HY (2013) Combining rabin cryptosystem and error correction codes to facilitate anonymous authentication with un-traceability for low- end devices. Comput Netw 57(14):2705–2717
16. Elia M, Schipani D (2013) On the rabin signature. J Discret Math Sci Cryptogr 16(6):367–378

17. Feng JB, Wu HC, Tsai CS, Chang YF, Chu YP (2008) Visual secret sharing for multiple secrets. Pattern Recogn 41(12):3572–3581
18. Gutub A, Al-Ghamdi M (2019) Image based steganography to facilitate improving counting-based secret sharing. 3D Research 10(1):6
19. Hayat U, Azam NA (2019) A novel image encryption scheme based on an elliptic curve. Signal Process 155:391–402
20. Hoffstein J, Pipher JC, Silverman JH, Silverman JH (2008) An introduction to mathematical cryptography. Springer, New York
21. Hou YC (2003) Visual cryptography for color images. Pattern Recognit 36(7):1619–1629
22. Kabirirad S, Eslami Z (2019) Improvement of (n, n)-multi-secret image sharing schemes based on boolean operations. J Inform Secur Appl 47:16–27
23. Kaya K, Selçuk AA (2007) Threshold cryptography based on Asmuth–Bloom secret sharing. Inform Sci 177(19):4148–4160
24. Kim C, Shin D, Leng L, Yang CN (2018a) Lossless data hiding for absolute moment block truncation coding using histogram modification. J Real-Time Image Proc 14(1):101–114
25. Kim C, Shin D, Leng L, Yang CN (2018b) Separable reversible data hiding in encrypted halftone image. Displays 55:71–79
26. Kim C, Yang CN, Leng L (2020) High-capacity data hiding for abtc-eq based compressed image. Electronics 9(4):644
27. Kurosawa K, Ogata W (1999) Efficient rabin-type digital signature scheme. Des Codes Crypt 16(1):53–64
28. Kurosawa K, Takagi T (2009) One-wayness equivalent to general factoring. IEEE Trans Inf Theory 55(9):4249–4262
29. Kurosawa K, Ito T, Takeuchi M (1988) Public key cryptosystem using a reciprocal number with the same intractability as factoring a large number. Cryptologia 12(4):225–233
30. Leng L, Zhang J, Khan K, Alghathbar K (2010) Dynamic weighted discrimination power analysis: A novel approach for face and palmprint recognition in dct domain. Acad J 05:2543–2554
31. Leng L, Li M, Kim C, Bi X (2017) Dual-source discrimination power analysis for multi-instance contactless palmprint recognition. Multimed Tools Appl 76(1):333–354
32. Liao X, Yin J, Chen M, Qin Z (2020) Adaptive payload distribution in multiple images steganography based on image texture features, ieee transactions on dependable and secure computing. IEEE Transactions on Dependable and Secure Computing
33. Lin CC, Tsai WH (2004) Secret image sharing with steganography and authentication. J Syst Softw 73(3):405–414
34. Lin PY, Chan CS (2010) Invertible secret image sharing with steganography. Pattern Recogn Lett 31(13):1887–1893
35. Liu YX, Yang CN, Chou YS, Wu SY, Sun QD (2018) Progressive (k, n) secret image sharing scheme with meaningful shadow images by gemd and rgemd. J Vis Commun Image Represent 55:766–777
36. Logeshwari R, Parvathy LR (2020) Generating logistic chaotic sequence using geometric pattern to decompose and recombine the pixel values. Multimedia Tools and Applications
37. Luo W, Huang F, Huang J (2010) Edge adaptive image steganography based on LSB matching revisited. IEEE Trans Inform Forens Secur 5(2):201–214
38. Luo Y, Yu J, Lai W, Liu L (2019) A novel chaotic image encryption algorithm based on improved baker map and logistic map. Multimed Tools Appl 78(15):22,023–22,043
39. Menezes AJ, Katz J, Van oorschot PC, Vanstone SA (1996) Handbook of applied cryptography. CRC Press, Boca Raton
40. Meng K, Miao F, Huang W, Xiong Y (2020) Threshold changeable secret sharing with secure secret reconstruction. Inform Process Lett 157:105,928
41. Mishra D, Sharma H, Sharma R, Kumar N (2017) A first cryptosystem for security of two-dimensional data. Fractals 25(01):1750,011
42. Naor M, Shamir A (1994) Visual cryptography. In: Workshop on the theory and application of of cryptographic techniques. Springer, New York, pp 1–12
43. Rabin MO (1979) Digitalized signatures and public-key functions as intractable as factorization. Tech. rep. Massachusetts Inst of Tech Cambridge lab for computer science
44. Sardar MK, Adhikari A (2020) Essential secret image sharing scheme with small and equal sized shadows. Signal Process Image Commun 87:115,923
45. Sarkar S, Kisku B, Misra S, Obaidat MS (2009) Chinese remainder theorem-based rsa-threshold cryptography in MANET using verifiable secret sharing scheme. In: 2009 IEEE International conference on wireless and mobile computing, Networking and Communications. IEEE, pp 258-262
46. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613

47. Sun S (2016) A novel edge based image steganography with 2k correction and huffman encoding. Inf Process Lett 116(2):93–99
48. Tang M, Hu J, Song W (2014) A high capacity image steganography using multi-layer embedding. Optik-Int J Light Elect Opt 125(15):3972–3976
49. Tripathi SK, Badiya S, Pandian KS, Gupta B, AlKhzaimi H (2020) Invertible secret sharing: Using meaningful shadows based on sorted indexed code. Optik 224:165,658
50. Tsai DS, Horng G, Chen TH, Huang YT (2009) A novel secret image sharing scheme for true-color images with size constraint. Inf Sci 179(19):3247–3254
51. Wang H, Wang S (2004) Cyber warfare: steganography vs. steganalysis. Commun ACM 47(10):76–82
52. Wu X, Yang CN (2019) Invertible secret image sharing with steganography and authentication for ambtc compressed images. Signal Process Image Commun 78:437–447
53. Wu X, Ou D, Liang Q, Sun W (2012) A user-friendly secret image sharing scheme with reversible steganography based on cellular automata. J Syst Softw 85(8):1852–1863
54. Wu X, Yang CN, Yang YY (2020) Sharing and hiding a secret image in color palette images with authentication. Multimed Tools Appl 79(35):25,657–25,677
55. Yang CN, Chen TS, Yu KH, Wang CC (2007) Improvements of image sharing with steganography and authentication. J Syst Softw 80(7):1070–1076
56. Yang CN, Ouyang JF, Harn L (2012) Steganography and authentication in image sharing without parity bits. Opt Commun 285(7):1725–1735
57. Yang CN, Chen CH, Cai SR (2016) Enhanced boolean-based multi secret image sharing scheme. J Syst Softw 116:22–34