Comparison of Encryption Techniques to Encrypt Private Parts of an Image

Nisha P. Shetty, Balachandra Muniyal, Rithish Reddy Kaithi, and Sarath Chandra Reddy Yemma

1 Introduction

Along with everyday advances in modern technology, the usage of image data among the numerous amount of systems and devices has increased. Today, systems like the IoTs, CCTVs, and drones having built-in imaging devices can obtain images of public and automatically store and share them. Thus, increasing use of the Internet by the public and the availability and sharing of public and private digital data have led industry experts and researchers to give special attention to information security. And there is a need to protect this private information from exponentially growing unauthorized access and attacks. Cryptography is about building, studying, and analyzing protocols that prevent third parties or the public from accessing private information. Main objective of cryptography techniques is to make the content of a message incomprehensible to unauthorized people. Every enciphering/encryption and deciphering/decryption process have two main aspects: the scheme of algorithm and the usage of keys for enciphering/encryption and deciphering/decryption. However, the security of the cryptographic process is maintained by the key used to encrypt and decrypt the information. There are two kinds of mechanisms in cryptography: Symmetric key cryptographic mechanism, in which equal key is used to encode and decode. In the asymmetric key cryptographic mechanism, two distinct keys are used to encrypt and decrypt. The algorithm for symmetric key is much faster, easier to implement, and requires less computing power compared to the asymmetric key algorithm. Steganography techniques are supported by concealing the existence of data by embedding the key message in another cover medium. A map displaying some kind of chaotic behavior is a chaotic map. A chaotic map is a difference equation describing a chaotic dynamical system which is discreet. A chaotic map is

N. P. Shetty (✉) · B. Muniyal · R. R. Kaithi · S. C. R. Yemma
Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
e-mail: Nisha.pshetty@manipal.edu

T. Sengodan et al. (eds.), *Advances in Electrical and Computer Technologies*,
Lecture Notes in Electrical Engineering 881,
https://doi.org/10.1007/978-981-19-1111-8_40

normally a discrete map that has sensitive dependence on initial conditions. Due to their appealing features like sensitivity to the initial conditions and random spreading out behavior, chaotic maps are adopted for several applications to protect the data. The primary target of this paper is to execute a picture encryption and decoding utilizing both methods and compare the experimental results and security analysis.

2 Related Work

Amal Hafsa et al. [3] propose an encryption scheme that includes the "advanced encryption standard (AES)" and "elliptic curve cryptography (ECC)" for encrypting medical images, combining the advantages of symmetric AES for the acceleration of encryption of data and asymmetric ECC for the secure exchange a symmetric session key. The analysis results prove the efficiency, speed, and better security of this algorithm. Arab et al. [4] propose a combination of both modified AES algorithm and chaotic sequence to develop a novel image encryption algorithm. Modified AES scheme is used to encrypt the original image using the encryption key, which is generated with the help of Arnold chaos sequence, and chaotic sequence produced round keys implementation. It is observed from the results from simulation that when small changes are introduced into the original image and cipher key, it results in a consequential change in the enciphered image making the original image inaccessible. Alsaffar et al. [5] compare "advanced encryption standard (AES)" and "Rivest-Shamir-Adleman (RSA)" algorithms for image encryption. Results from tests reveal that image encryption quality is better with AES algorithm, with uniform distribution of pixels in histogram. Moreover, coefficient of correlation is closer to 0 for AES algorithm. The overall results of this study conclude that the AES algorithm is a better algorithm than RSA algorithm for image encryption. Farah et al. [6], to enhance the performance of encryption algorithms, proposed a novel hybrid chaotic map and a distinct way of utilizing optimization technique. In the proposed algorithm, S -boxes based on their respective nonlinearity score are generated accordingly using chaotic Jaya optimization algorithm to carry out a new optimized S-box dependent substitution phase. Results from security analysis reveal that the proposed encryption scheme is resistant to various unauthorized attacks. Ping et al. [7] proposed discrete Henon map, attaining a novel two-point diffusion techniques. The key stream for substitution is generated and made based on the original image to obtain the greater capability of showing resistance against chosen-plaintext attack or known-plaintext attack. The results from security analysis reveal that this encryption scheme provides a superior security, meanwhile, it is observed that faster encryption speed is achieved compared to various image encryption schemes from the time complexity analysis. Shadangi et al. [8] propose a novel CBC-AES algorithm to encrypt images based on Arnold scrambling having several encryption levels. In this method, at first, the original image is shifted circularly and Arnold scrambled together with a bit-wise shuffle operation, after which each bit is complemented. Finally, CBC mode of AES encryption scheme was employed to attain the resulting cipher image.

Furthermore, it is evident from the results of security analysis that this algorithm has ability of resisting various attacks like differential attacks, statistical attacks, and entropy attacks. Shaktawat et al. [9] propose a hybrid approach for image encryption by combining AES, a standard cryptography algorithm, along with splitting and block permutation. Comparison of various parameters with and without block permutation confirms the superiority of the proposed method with regard to better results after using splitting and permutation functions. Wang et al. [10] proposed chaotic encryption of image with a novel one-time pad design which is based upon the multiple mixed combinations of hash functions mixed and cyclic-shift function. By using both the data from the original image and the chaotic sequences, which are calculated using hash algorithms such as SHA1 and MD5 hash, the initial value is generated. The logistic map and the nonlinear equations generate the scrambling sequences. Experimental results and security analysis prove better security provided by this scheme and are protected from common attacks. Hua et al. [11] present a "two-dimensional (2D) logistic-sine-coupling map (LSCM)". Furthermore, they propose an algorithm to encrypt image based on 2D-LSCM (LSCM-IEA) to adopt classical confusion-diffusion structure using the proposed 2D-LSCM. To interchange pixels of image to various rows and columns, a permutation algorithm is devised and to impart changes of original image to encrypted image, a diffusion algorithm is designed. The results from analysis reveal that better security performance than various algorithms is obtained by LSCM-IEA. Hua et al. [12] proposed an encryption scheme that uses highly efficient scrambling to split adjoining pixels and employing substitution in a random order to impart a small variation in the original image to each and every pixel of the encrypted image. The security analysis reveals that security provided by this encryption scheme is better than various modern image encryption algorithms.

3 Methods Used

3.1 Combination of AES, RSA, LSB

- In December 2001, The National Institute of Standards and Technology (NIST) published "advanced encryption standard (AES)" as FIPS 197 (Federal Information Processing Standard). AES is a symmetric block cipher. The algorithm was developed by Joan Daemen and Vincent Rijmen, Belgian Researchers. The AES algorithm is easy to implement and uses keys of different length (128, 192, or 256 bits) according to number of rounds (10, 12, or 14, respectively) used in the implementation of the algorithm.

  Each round of AES has 4 steps:

1. SubByte:
   It uses the substitution box (s-box) to substitute the values of each byte. Each byte is interpreted as 2 digit hexadecimal value.

2. ShiftRows:
   The permutation step of the method, each row of the matrix is shifted cyclically to the left.
3. MixColumns:
   Each column of the matrix is multiplied by a constant to transform it into a new column.
4. AddRoundKey:
   The step that uses the key, transforms each column by adding it to the key.

The process of encryption is done after the completing all the rounds (10 in our case).

- RSA algorithm named after its publishers Ron Rivest, Adi Shamir, and Leonard Adleman. It is a public key, asymmetric encryption algorithm where only the receiver has the private key. Both sender and receiver have the value of "n". The public key is {e, n} which is used for encryption, and the private key is {d, n} used for decryption.

  Where
  $n = p^*q$, (p and q are prime numbers which are selected and kept private).

e, chosen such that e,$(p-1)*(q-1)$ are co-primes
$d = (\mathrm{e}^{-1}) \bmod ((p-1)*(q-1))$
Encryption of plaintext M:

$$C = M^{\mathrm{e}} \bmod n \tag{1}$$

Decryption of cipher text C:

$$M = C^{d} \bmod n \tag{2}$$

- Steganography is a method of hiding information in the ordinary plain text or image to avoid detection. LSB method is a steganography method in which the message to be hidden replaces the least significant bit of the pixel values.

The combination of these methods is to cipher the 128 bit key, by which the image is encrypted in AES encryption, using RSA algorithm and places the obtained ciphertext into the image with LSB method.

3.1.1 Encryption

The values of the image are traversed pixel by pixel and in doing so it encrypts each value and append them to the new data array (new_data). The method uses ciphertext feedback (CFB) mode of operation. The key of length 16 bytes is produced randomly, and the initialization vector (IV) of size 16 bytes is to be used.

- As each pixel contains r, g, b values are converted into bytes for the encryption.

```
cfb_cipher = AES.new (key, AES.MODE_CFB, IV)
k = bytes (pixel [0], pixel [1], pixel [2]))
enc_data = cfb_cipher.encrypt(k)
```

The encrypted value is in bytes. So it is converted back into rgb values and places in new_data.

```
Val = rgb (enc_data)
data_b[ind] = Val
```

Finally, a new encrypted image is formed from new_data.

- The key used is converted into hexadecimal and then ciphered and converted into bits for LSB method. The values of n, e, d are declared ahead of implementing the method.

```
num=key.hex().
X = (num[i] **e) %n; i = 0 to length of num.
Message = binary(X) (converting it into binary)
```

- A constant string is appended to the message to specify the end in decryption process. Each bit of the message is placed in the least significant bit of the pixel value of the encrypted image formed by AES method.

```
rgb = rgb[:-1]+ message[i] ,where i = 0 to length of message
```

, rgb is value of a pixel.

The success of LSB method does not show changes of the encrypted image, before and after the implementation of the process, to the naked eye.

3.1.2 Decryption

The decryption process is similar to encryption process except we retrieve the key from the encrypted image and use it to decrypt it to get original image. Same key and initialization vector are to be used to get the original image.

- The below step is repeated until the constant string is encountered.

```
message += rgb[-1]
```

- The message is converted into hexadecimal for deciphering with RSA. Converting Y into bytes gives the key used to encrypt the image.

```
message = hex(message)
Y += (message[i] **d) %n; i = 0 to length of message.
Key = bytes(Y)
```

- Each r, g, b value of the pixel is converted into bytes and then decrypted to get the original value at the pixel.

```
cfb_cipher = AES.new (key, AES.MODE_CFB, IV)
```

```
k = bytes (pixel [0], pixel [1], pixel [2]))
enc_data = cfb_cipher.decrypt(k)
```

The encrypted value is in bytes. So it is converted back into rgb values and places in new_data.

```
Val = rgb (enc_data)
data_b[ind]=Val
```

- Finally, the image formed by the new_data is the decrypted image.

3.2 Hash Function (Chaotic Method)

Any function that converts an input of variable length (generally a large value) into an output of fixed small range value is called hash function. In this method of encryption, we use SHA-2 (a 1D hash) algorithm to create a 2D mask. The SHA-2 operation can be divided into two parts:

- Pre-processing:
  It includes padding of the original message of length $<2^{128}$ to make the length a multiple of 1024 bits along with the length of original message. The padded message is then decomposed into blocks of 1024 bits each. Initial values that are used in hash generation are initialized.
- Hash computation:
  The final message digest is formed after a number of rounds of hash value that is produced using the padded message from the pre-processing step. It uses functions, word logics, and other operations to get hash functions.

  This method uses substitution-diffusion type hash-based image encryption. This has four stages, two of which involve both substitution and diffusion while the other two involves only the diffusion process. In the process substitution, each pixel value is interpreted as hexadecimal value and is converted using s-box of the AES encryption method.

  In this algorithm, the image is divided into 4 subparts (Im.1, Im.2, Im.3, Im.4 each of 128 × 128) for both encryption and decryption processes. The idea is to encrypt half on the image with the remaining half.

  Description of the methods used in the algorithm:

I. Substitution of subpart Im.X using S-box of AES algorithm(sbox(Im.X))

II. XOR of columns of each row of subpart Im.X(XCR(Im.X)):
  For a subpart, XOR all the *r*, *g*, *b* values with values corresponding to them in the column in row *i*, where $i = 1, 2... . , 127, 128$. The result is a matrix of size 128 × 1. The matrix is horizontally concatenated 128 times to create a matrix of 128 × 128.

III. XOR of rows of each column of subpart Im.X(XRC (Im.X)):
  For a subpart, XOR all the *r*, *g*, *b* values with values corresponding to them in the row in column *i*, where $i = 1, 2…., 127, 128$. The result is a matrix

of size 1 × 128. The matrix is vertically concatenated 128 times to create a matrix of 128 × 128.

IV. Subparts hashing (Hash (Im.X, Im.Y)):

Subparts Im.X and Im.Y together create a matrix of size 128 × 256, say H. Each row of H is then divided to form four arrays of size 124 bytes. To these arrays, four keys of 1 byte each will be appended to form 128 byte array. For each row, the 4 sub-arrays and keys that are appended are shown below:

$$\begin{aligned} \text{SA1} &= \text{H}[1:124] + \text{Keys}(1\text{ to }4) \\ \text{SA2} &= \text{H}[51:174] + \text{Keys}(5\text{ to }8) \\ \text{SA3} &= \text{H}[100:223] + \text{Keys}(9\text{ to }12) \\ \text{SA4} &= \text{H}[132:255] + \text{Keys}(13\text{ to }16) \end{aligned}$$

The resultant of hash function is 64 bytes for each sub-array. These results together, 256 bytes, are the output of hash function of each row. By the end of all the rows, a new matrix of size 128 × 256 is created.

V. Substitution of subpart Im.X according to AES algorithm's inverse sbox (InvSbox (Im.X))

3.2.1 Encryption

The height and width of the image should be 256, else they are to be adjusted. In this algorithm, 16 keys each of 8 bits are used. The image is divided into 4 subparts, 128 × 128 pixels each. In the method, lower half's information is used to encrypt the upper half and vice-versa.

Remarks In each step, Im.X_{new} is considered as subpart that is a result obtained by using $\text{Im.X}_{\text{old.}}$ (Im.X_{old} is the subpart that is about to be used in present step, and Im.X_{new} is the result of the present step).

Step 1:

Choose the secret keys (total 16) to perform method IV. Substitute subparts Im.1 and Im.2 according to sbox of AES. Calculate XRC (Im.3), XCR (Im.3), XRC(Im.4), XCR (Im.4), and hash (Im.3 and Im.4). The results of the step are Im.1_{new}, Im.2_{new}.

$$\text{Im.1}_{\text{new}} = \text{sbox}\,(\text{Im.1}_{\text{old}}) \oplus \text{XRC}\,(\text{Im.3}) \oplus \text{XCR}\,(\text{Im.3}) \oplus \text{Hash}\,(\text{Im.3},\ \text{Im.4}). \tag{3}$$

$$\text{Im.2}_{\text{new}} = \text{sbox}(\text{Im.2}_{\text{old}}) \oplus \text{XRC}\,(\text{Im.4}) \oplus \text{XCR}\,(\text{Im.4}) \oplus \text{Hash}(\text{Im.3},\ \text{Im.4}). \tag{4}$$

Remarks The result of the hash is 128 × 256, and the columns from 1 to 128 are used in step (1) and from 129 to 256 are used in step (2).

Step 2:

Substitute subparts Im.3 and Im.4 according to S-box of AES. Calculate XRC (Im.1), XCR (Im.1), XRC (Im.2), XCR (Im.2), and hash (Im.1 and Im.2). The results of the step are Im.3_{new}, Im.4_{new}.

$$\text{Im.3}_{\text{new}} = \text{sbox}\,(\text{Im.3}_{\text{old}}) \oplus \text{XRC}\,(\text{Im.1}) \oplus \text{XCR}\,(\text{Im.1}) \oplus \text{Hash}\,(\text{Im.1},\ \text{Im.2}). \tag{5}$$

$$\text{Im.4}_{\text{new}} = \text{sbox}\,(\text{Im.4}_{\text{old}}) \oplus \text{XRC}\,(\text{Im.2}) \oplus \text{XCR}\,(\text{Im.2}) \oplus \text{Hash}\,(\text{Im.1},\ \text{Im.2}). \tag{6}$$

Remarks The result of the hash is 128×256, and the columns from 1 to 128 are used in step (1) and from 129 to 256 are used in step (2).

Step 3:

Calculate XRC (Im.2), XCR (Im.2), XRC (Im.4), XCR (Im.4). The results of the step are Im.1_{new}, Im.3_{new}.

$$\text{Im.1}_{\text{new}} = \text{Im.1}_{\text{old}} \oplus \text{XRC}\,(\text{Im.2}) \oplus \text{XCR}\,(\text{Im.4}) \tag{7}$$

$$\text{Im.3}_{\text{new}} = \text{Im.3}_{\text{old}} \oplus \text{XRC}\,(\text{Im.4}) \oplus \text{XCR}\,(\text{Im.2}) \tag{8}$$

Step 4:

Calculate XRC (Im.1), XCR (Im.1), XRC (Im.3), XCR (Im.3). The results of the step are Im.2_{new}, Im.4_{new}.

$$\text{Im.2}_{\text{new}} = \text{Im.2}_{\text{old}} \oplus \text{XRC}\,(\text{Im.1}) \oplus \text{XCR}\,(\text{Im.3}) \tag{9}$$

$$\text{Im.4}_{\text{new}} = \text{Im.4}_{\text{old}} \oplus \text{XRC}\,(\text{Im.3}) \oplus \text{XCR}\,(\text{Im.1}) \tag{10}$$

All the 4 subparts are then appended accordingly to form a 256×256 encrypted image.

3.2.2 Decryption

The decryption process is similar to encryption process except the order is reversed, and in place of S-box, we use InvSbox of AES algorithm. The encrypted image is divided into 4 subparts.

Step 1: This step is same as the step 4 of encryption process.
Step 2: This step is same as the step 3 of encryption process.

Step 3: Calculate XRC (Im.1), XCR (Im.1), XRC (Im.2), XCR (Im.2), and hash(Im.1 and Im.2). The result of the step is Im.3_{new}, Im.4_{new}.

$$\text{Im.3}_{\text{new}} = \text{InvSbox}\left(\text{Im.3}_{\text{old}} \oplus \text{XRC}\left(\text{Im.1}\right) \oplus \text{XCR}\left(\text{Im.1}\right) \oplus \text{Hash}\left(\text{Im.1},\ \text{Im.2}\right)\right) \tag{11}$$

$$\text{Im.4}_{\text{new}} = \text{InvSbox}\left(\text{Im.4}_{\text{old}} \oplus \text{XRC}\left(\text{Im.2}\right) \oplus \text{XCR}\left(\text{Im.2}\right) \oplus \text{Hash}\left(\text{Im.1},\ \text{Im.2}\right)\right) \tag{12}$$

Remarks The result of the hash is 128 × 256, and the columns from 1 to 128 are used in step (9) and from 129 to 256 are used in step (10).

Step 4:

Calculate XRC (Im.3), XCR (Im.3), XRC (Im.4), XCR (Im.4), and hash(Im.3 and Im.4). The result of the step is Im.1_{new}, Im.2_{new}.

$$\text{Im.1}_{\text{new}} = \text{InvSbox}\left(\text{Im.1}_{\text{old}} \oplus \text{XRC}\left(\text{Im.3}\right) \oplus \text{XCR}\left(\text{Im.3}\right) \oplus \text{Hash}\left(\text{Im.3},\ \text{Im.4}\right)\right) \tag{13}$$

$$\text{Im.2}_{\text{new}} = \text{InvSbox}\left(\text{Im.2}_{\text{old}} \oplus \text{XRC}\left(\text{Im.4}\right) \oplus \text{XCR}\left(\text{Im.4}\right) \oplus \text{Hash}\left(\text{Im.3},\ \text{Im.4}\right)\right) \tag{14}$$

Remarks The result of the hash is 128 × 256, and the columns from 1 to 128 are used in step (11) and from 129 to 256 are used in step (12).

All the 4 subparts are then appended accordingly to form a 256 × 256 decrypted image.

Below figures show the encrypted images using both algorithms (Fig. 1):

4 Security Test and Comparative Study

4.1 Statistical Analysis

Statistical analysis based on the histograms of original and respective cipher images, the correlation of adjacent pixels in the cipher images and the coefficient of correlation for original images and their respective cipher images are taken into consideration to examine the robustness of the image encryption algorithms.

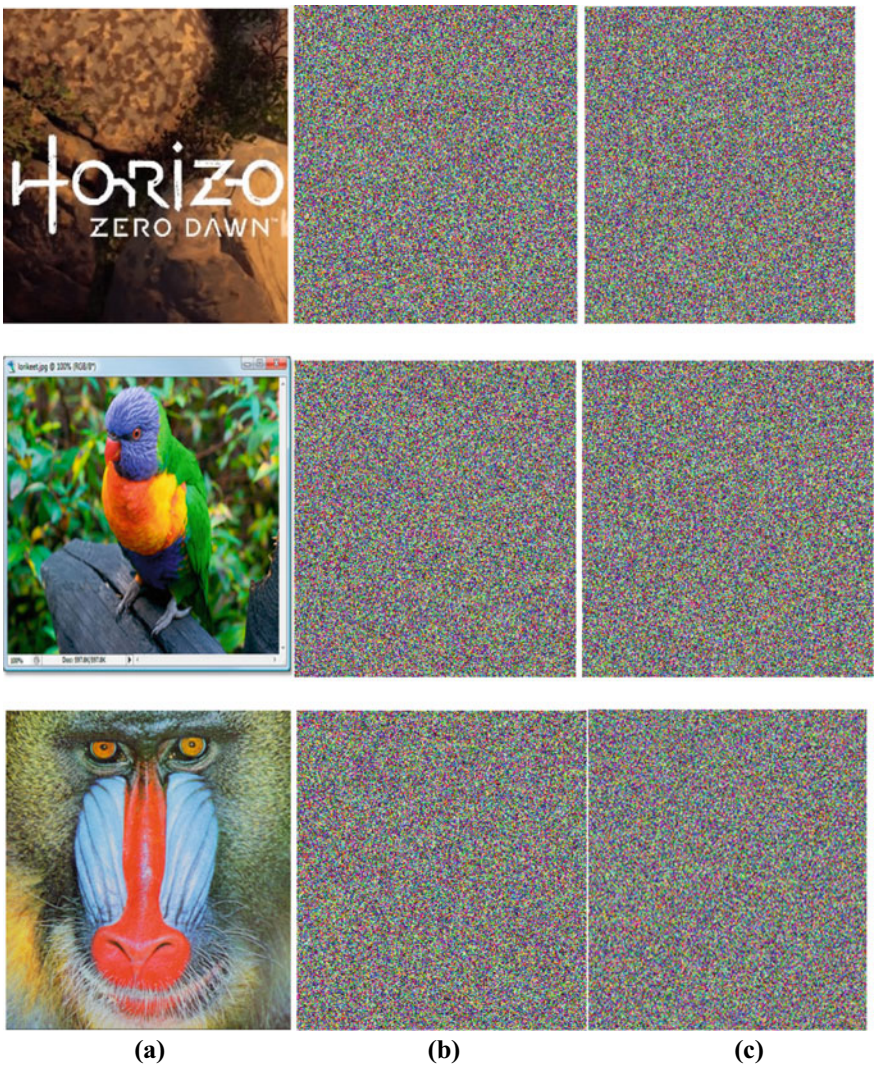


Fig. 1 **a** Plain original images **b** corresponding encrypted images using AES-RSA-LSB algorithm **c** corresponding hash-encrypted images

4.1.1 Histogram Analysis

An image-histogram is a histogram that acts as a graphical illustration of image pixel distribution in a digital image by indicating the no. of pixels at each tonal intensity level.

In Fig. 2, histogram of the original image 1(HorizonZero) of size 250 × 250 demonstrates that the histogram of original image has particular pattern of r, g, and b components.

Figure 3 shows us the corresponding cipher image, encrypted using AES-RSA-LSB method, and its respective histogram.

Figure 4 shows us the hash-encrypted image of original image HorizonZero and its respective histogram.

Figure 5: The histogram of the original image 2 of size 250 × 250 demonstrates that the histogram of original image has particular pattern of r, g, and b components.

Figure 6 shows us the corresponding cipher image, encrypted using AES-RSA-LSB method, and its respective histogram.

Figure 7 shows us the hash-encrypted image of original image 2 and its respective histogram.

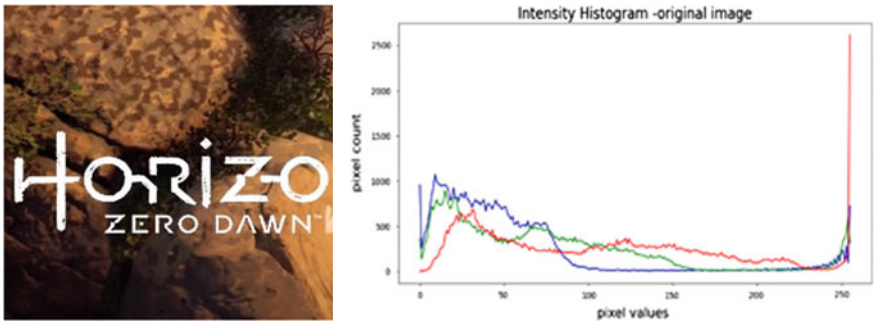


Fig. 2 **a** Original image HorizonZero, **b** histogram of original image HorizonZero

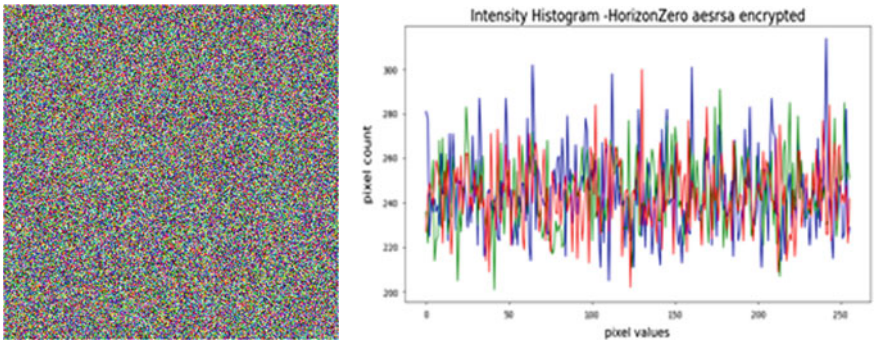


Fig. 3 **a** Encrypted image of original image HorizonZero using AES-RSA-LSB method, **b** histogram of (**a**)

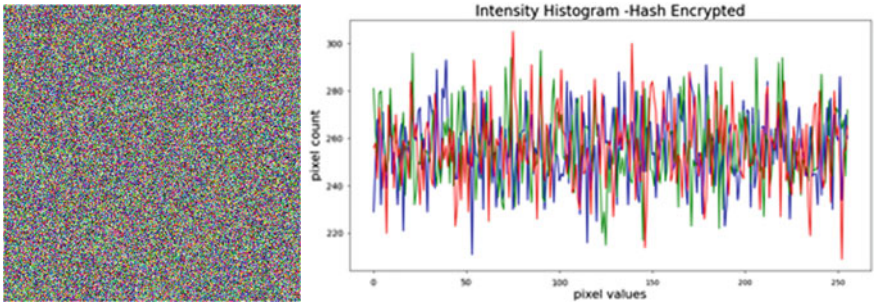


Fig. 4 **a** Hash-encrypted image of original image HorizonZero, **b** histogram of Fig. 4a

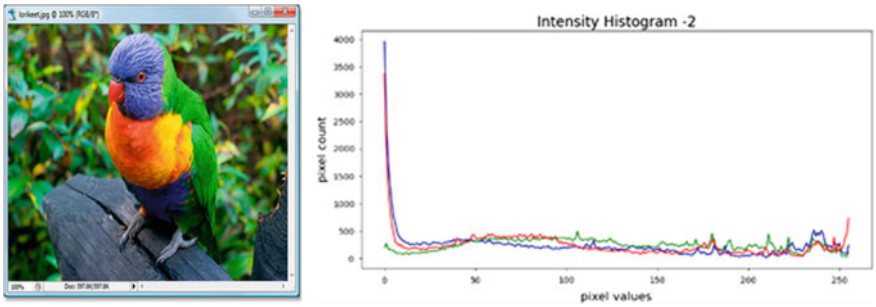


Fig. 5 **a** Original image 2, **b** histogram of original image 2

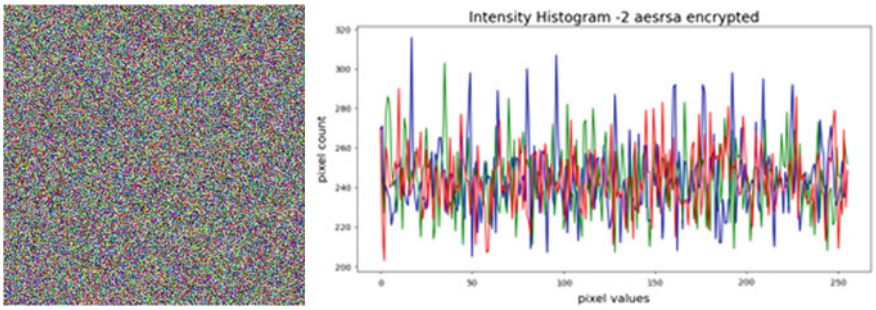
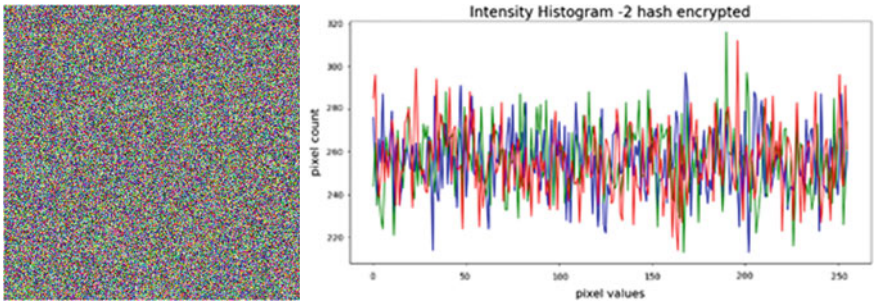


Fig. 6 **a** Encrypted image of original image 2 using AES-RSA-LSB method, **b** histogram of Fig. 6a



(a) (b)

Fig. 7 **a** Hash-encrypted image of original image 2, **b** histogram of Fig. 7a

It should be noted that all pixels are uniformly distributed in the cipher image, therefore making cryptanalysis more difficult. This test is conducted for both algorithms using both images and more satisfactory results were obtained with the hash algorithm.

4.1.2 Correlation Coefficient Analysis

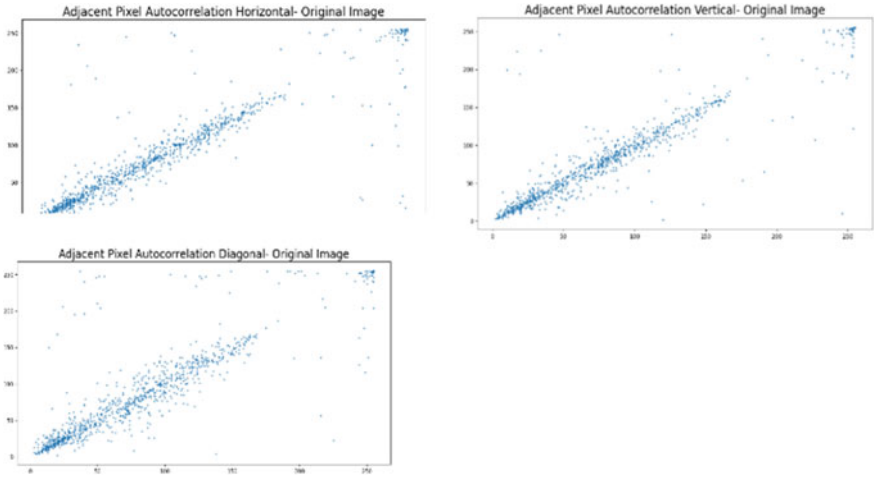
Correlation is a process of determining the probability that there exists a linear relation among two weighted values. Karl Pearson described the Pearson product-moment coefficient of correlation, *r*, in 1895. In image processing, pattern recognition, and statistical analysis, the Pearson's coefficient of correlation, r, was widely used as a correlation measure. The Pearson's coefficient of correlation for digital images is defined as

$$r_{XY} = \frac{\sum (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum (X_i - \overline{X})^2 \sum (Y_i - \overline{Y})^2}} \tag{13}$$

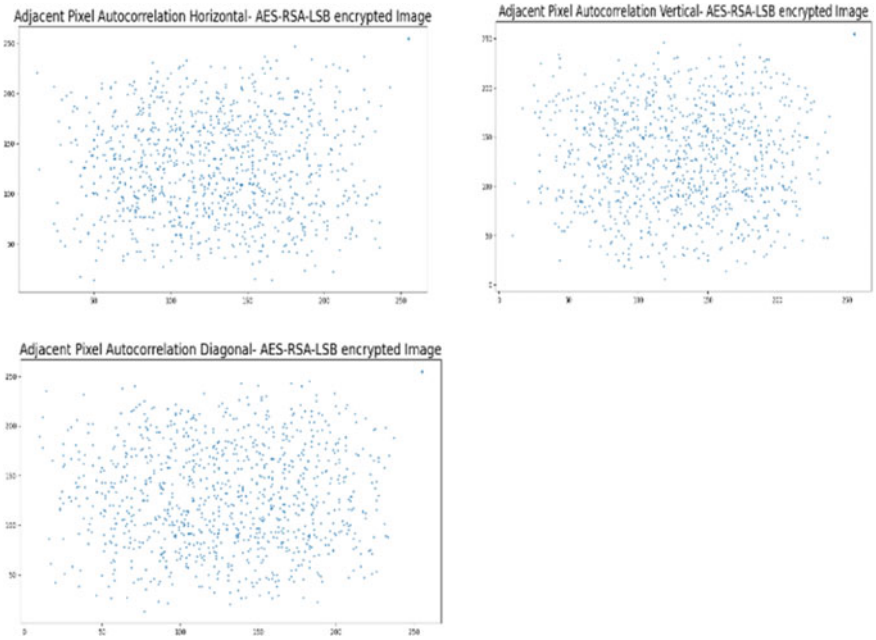
It calculates the relation between two adjacent pixel values. If they are absolutely identical, coefficient of correlation has the value $r = 1$. They are absolutely uncorrelated if $r = 0$ and are absolutely anti-correlated if the value of $r = -1$. Horizontal correlation, vertical correlation, and diagonal correlation plotted between adjacent pixels of both the original image HorizonZero and images encrypted with encryption algorithm using AES-RSA-LSB, hash function are shown in the Fig. 8.

From Fig. 8a, it is clearly observable that there is a certain pattern in the correlation plot of adjacent pixels in original image HorizonZero, but Fig. 8b and c show us that there is uniform pixel distribution in encrypted images of both algorithms.

Horizontal correlation, vertical correlation, and diagonal correlation plotted between adjacent pixels of both the original image 2 and images encrypted with encryption algorithm using AES-RSA-LSB, hash function are shown in the Fig. 9.

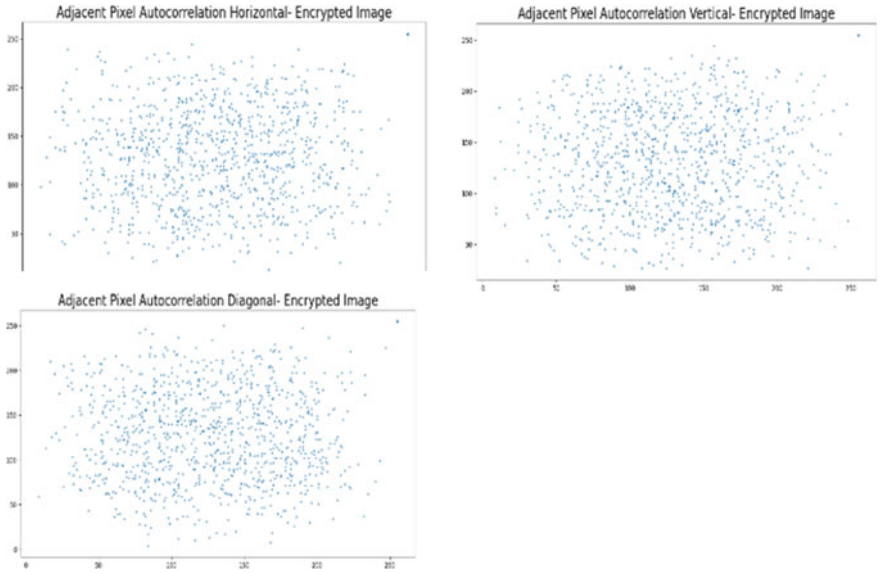


(a)



(b)

Fig. 8 **a** Distribution of pixels in original image HorizonZero, **b** distribution of pixels in encrypted image using AES-RSA-LSB method, **c** distribution of pixels in hash-encrypted image



(c)

Fig. 8 (continued)

From Fig. 9a, it is clearly observable that there is a certain pattern in the correlation plot of adjacent pixels in original image 2, but Fig. 9b and c show us that there is uniform pixel distribution in encrypted images of both algorithms.

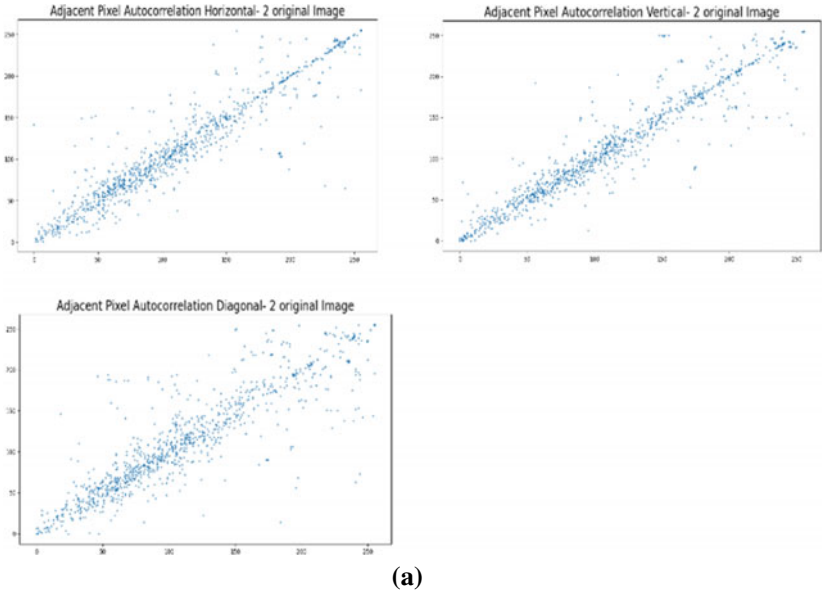
Tables 1 and 2 show us that the values of correlation coefficient of encrypted images using both algorithms are closer to 0 than that of original images, but it is notable that correlation coefficients of hash-encrypted image are more precise to 0.

4.2 Information Entropy Analysis

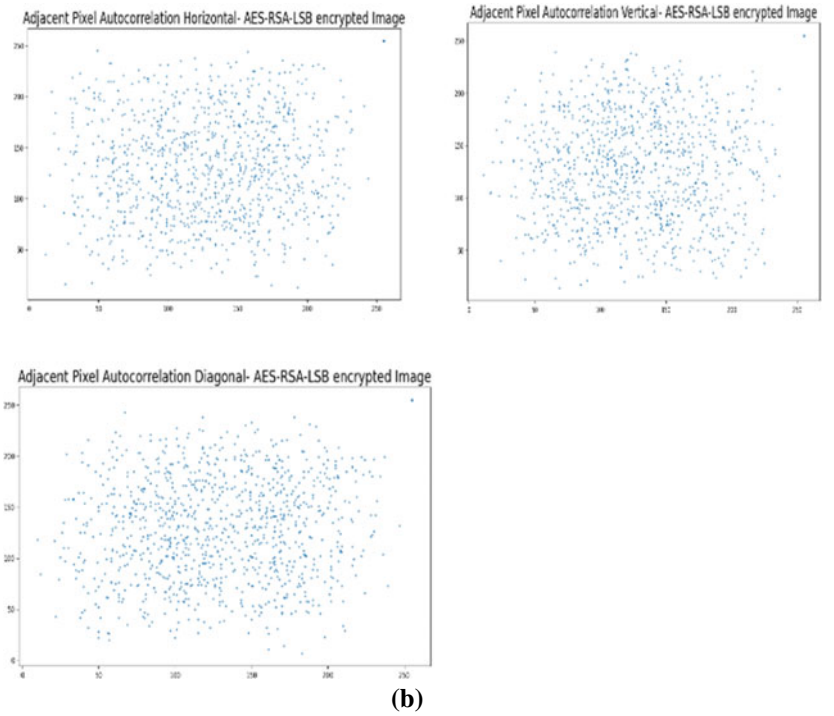
Entropy is often used for the calculation of the amount of information within an image and the randomness that is indicated by the image's texture and the amount of information within an image. The entropy $H(s)$ is defined as

$$H(S) = -\sum_{i=1}^{m} p_i \log_2 p_i \tag{14}$$

where p_i stands for the probability of message. The ideal value of information entropy, when an image is encrypted, is considered to be 8, if the value is lower than this, then there remains a definite degree of certainty that its security might be compromised (Table 3).

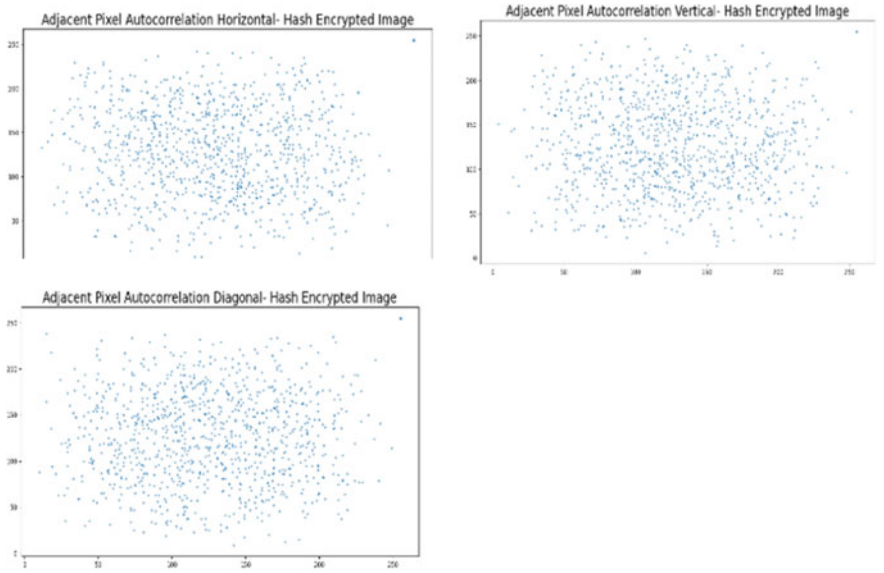


(a)



(b)

Fig. 9 **a** Distribution of pixels in original image 2, **b** distribution of pixels in encrypted image using AES-RSA-LSB method, **c** distribution of pixels in hash-encrypted image



(c)

Fig. 9 (continued)

4.3 Encryption Quality–PSNR, NPCR, UACI Analysis

PSNR Analysis:

"Peak signal-to-noise ratio (PSNR)" can be defined as the ratio between the original image and its respective cipher image (Table 4).

$$\begin{aligned} \text{PSNR} &= 10\log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right) \\ \text{MSE} &= \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,\ j) - K(i,\ j)]^2 \end{aligned} \tag{15}$$

MAX_I = Maximum value of pixel in an original image

m = Total number of rows in an original image

n = Total number of columns in an original image.

The greater the value of PSNR, the more is the resemblance of cipher image and the original. Normally, a greater PSNR value should be able to correlate to a better quality image. Encryption scheme is considered good if the PSNR value is as low as possible. Without the learning of secret key, lower PSNR values can reason the challenge in obtaining the original image from the encrypted image.

NPCR and UACI analysis:

By analyzing both the NPCR and the UACI values for images encrypted with the AES-RS-LSB method and hash function, the evaluation of ability to show resistance against a chosen plain text attack (differential attack) was done. NPCR defines

Table 1 Coefficient of correlation of adjacent pixels of original image HorizonZero and 2 and cipher images using AES-RSA-LSB method (horizontal, vertical, and diagonal)

Image name	Horizontal correlation		Vertical correlation		Horizontal correlation	
	Original image	Enciphered image	Original image	Enciphered image	Original image	Enciphered image
HorizonZero	0.9214918184319467	0.013909011716219577	0.9347604292095957	0.011044511955837444	0.8768273314184477	0.024230078775364844
2	0.9276754566366184	0.013302685540301992	0.8692175319999583	0.011286853264362157	0.9275945654901759	0.023489688715665452

Table 2 Coefficient of correlation of adjacent pixels of original image HorizonZero and 2 and hash-encrypted images (horizontal, vertical, and diagonal)

Image name	Horizontal correlation		Vertical correlation		Diagonal correlation	
	Original image	Enciphered image	Original image	Enciphered image	Original image	Enciphered image
HorizonZero	0.9214918184319467	0.01155419667017823	0.9347604292095957	0.013058796261792245	0.8768273314184477	0.023844368140612762
2	0.9276754566366184	0.006025497419932511	0.9275945654901759	0.010301019107040054	0.8692175319999583	0.019655715741470316

Table 3 Shows the entropy values for original image and cipher images

Image name	Original image	Entropy with algorithm 1	Entropy with algorithm 2
HorizonZero	7.4924423342698985	7.998747402442854	7.9990382671485065
2	6.599263778679899	7.998825871300561	6.786805859225379
Baboon	7.692590923186865	7.998818479848452	7.998972035956279

the variation rate of pixels between original image and cipher image, and UACI defines the average change in intensity of the original image and the cipher image. There will be better ability to prove resistance against a chosen plain text attack if the original image is highly modified when compared to the cipher image. Even a small modification created in original image results in observable change. Significant correspondence between original image and cipher image can be found using this method.

"Number of pixel change rate (NPCR)" is defined as

$$\text{NCPR} = \frac{\sum_{i,\,j} D(i,\ j)}{M \times N} \times 100\%. \tag{16}$$

"Unified average change in intensity (UACI)" is defined as

$$\text{UACI} = \frac{1}{M \times N}\left[\sum_{i,\,j}\left[\frac{[I(i,\ j) - K(i,\ j)]}{255}\right]\right] \times 100\% \tag{17}$$

$$D(i,\ j) = \begin{cases} 0 & \text{if } I(i,\ j) = K(i,\ j) \\ 1 & \text{otherwise} \end{cases}.$$

If the NPCR value is more than 99%, then the analysis is positive. The greater the values of NPCR and the UACI, the superior the proposed algorithm is (Table 5).

5 Conclusions

We introduced image encryption algorithms that are based on AES, RSA and LSB methods, and hash function (SHA-2(512)). First algorithm uses RSA algorithm to encrypt the generated keys, then conceals the encrypted key using LSB steganography in image encrypted using AES algorithm. The second algorithm uses SHA-2, a 1D hash algorithm, to create a 2D mask to encrypt the image. The method is a substitution-diffusion type hash-based image encryption. Results from the conducted experiments acknowledge that encrypted images using both the suggested methods generate histograms with uniform distribution in pixel. The pixels are more uniformly distributed in the hash-encrypted images. Coefficient of correlation of adjacent pixels

Table 4 Shows the results of PSNR analysis

Image name	AES-RSA-LSB method		Hash encryption	
	PSNR	PSNR(db)	PSNR	PSNR(db)
HorizonZero	7.197269826170008	27.91007489488519	7.716409883513068	27.886869583915008
2	7.413889553591595	27.897974289640757	7.998875509097952	27.901030827344787
Baboon	8.95359387440058	27.90293351995701	9.775708777524772	27.898571899987957

Table 5 Measurement of encryption quality

Image	Encryption with AES-RSA-LSB		Encryption with hash	
	NPCR	UACI	NPCR	UACI
HorizonZero	99.60906666666665	33.43224470588695	99.58826666666668	33.30574431373036
2	99.60640000000001	35.432288627456124	99.60106666666665	35.59527529412285
Baboon	99.6032	29.532580392163016	99.60053333333333	29.567472941182693

in enciphered images is decreased greatly when compared to original images. Moreover, values of correlation coefficient of encrypted images using both algorithms are closer to 0 than that of original images, but it is notable that the coefficients of correlation for hash-encrypted images are more precise to 0. The suggested algorithms have proven that the encrypted pictures have information entropy of precise to 8, which is ideal. Information entropy of hash-encrypted images HorizonZero and baboon is closer to 8. These results prove robustness of both algorithms, also proving better robustness of hash function. The PSNR values are less AES-RSA-LSB algorithm compared to hash function. The NPCR analysis shows both algorithms give NPCR values above 99%, although they are close AES-RSA-LSB algorithm gives a better edge in results and UACI analysis yield better results in both the algorithms, but hash function gives slightly better results, which conclude superior quality and security of both the algorithms.

References

1. Moumen A, Sissaoui H (2017) Images encryption method using stenographic LSB method, AES and RSA algorithm. Nonlinear Eng 6(1):53–59
2. Seyedzade SM, Atani RE, Mirzakuchaki S (2010) A novel image encryption algorithm based on hash function. In: Iranian conference on machine vision and image processing. pp 5941167
3. Hafsa A, Sghaier A, Malek J, Machhout M (2021) Image encryption method based on improved ECC and modified AES algorithm. Multimedia Tools Appl 80:19769–19801
4. Arab A, Rostami MJ, Ghavami B (2019) An image encryption method based on chaos system and AES algorithm. J Supercomput 75(10):6663–6682
5. Alsaffar DM et al. (2020) Image encryption based on AES and RSA algorithms. In: 2020 3rd international conference on computer applications information security (ICCAIS), pp 1–5
6. Farah MB, Farah A, Farah T (2019) An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. Nonlinear Dyn 8:1–24
7. Ping P, Xu F, Mao Y, Wang Z (2017) Designing permutation substitution image encryption networks with Henon map. Neurocomputing. 1–17
8. Shadangi V, Choudhary SK, Patro KAK, Acharya B (2017) Novel Arnold scrambling based CBC-AES image encryption. Int J Control Theory Appl 10:93–105
9. Shaktawat VAR, Rs S, Lakshmi N, Panwar A (2020) A hybrid technique of combining AES algorithm with block permutation for image encryption. Rel Theory Appl 15(1):15
10. Wang X, Zhu X, Wu X, Zhang Y (2018) Image encryption algorithm based on multiple mixed hash functions and cyclic shift. Opt Lasers Eng 107:370–379
11. Hua Z, Jin F, Xu B, Huang H (2018) 2D logistic-sine-coupling map for image encryption. Signal Process 149:148–161
12. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. Inf Sci 480(1):403–419