An element-wise generalized singular value decomposition (GSVD) based image encryption algorithm

Suresh Babu V, V R Vijaykumar, T Gopalakrishnan

Christ The King Engineering College, Coimbatore, Tamil Nadu
Anna University, Regional Campus, Coimbatore, Tamil Nadu
Karpagam College of Engineering, Coimbatore, Tamil Nadu

sureshvece@gmail.com, vr.vijaykumar@gmail.com

Abstract. The image encryption algorithm aids a secure data transmission. In our proposed method, we use the generalized singular value decomposition (GSVD) as the encryption tool. Here, the secret image will be decomposed with respect to the key and generates the ciphered text. Further, to improve its security, the Arnold transforms and the XOR operation is used respectively. The region-wise applied GSVD reduce the time consumption of the conventional image encryption methods. Here, the GSVD decomposition is element-wise, so, the algorithm is more robust. The generated cipher text is transmitted and the key is generated with the combination of the key used to decompose the secret image pixel in GSVD module, the Arnold transform key and the XOR operator key. In receiver end, the ciphered image is synthesized with the key received in a separate channel. This proposed model is lossless encryption method.

1. Introduction
For past few decades, we have witnessed the significance of the digital data security. The amount of usage of digital data is increased rapidly in last few years. Even the usage among common people is increased gradually. The social media, the government policies and the advantages in use of digital data played a vital role in the increase of the use of digital data. The usage of digital image and video holds the significant share on the general digital data usage. The important security tool in digital data transmission is encryption. There are many traditional encryption algorithms. They maintained the best efficiency all over the time. But still, we have the demand for new, different, secure, fast and cheap image encryption methods. This demand still drives the researcher’s attention towards encryption schemes. The most famous real-time existing encryption scheme is data encryption standard (DES) and advanced encryption standard (AES) [1],[2]. These schemes are based on many rounds of confusion and diffusion algorithms [3]. But, recently, the encryption schemes have many types of classification based on its application, data, technique used for the encryption and so on. Among those methods a familiar method is chaos based encryption schemes [4]. There are many algorithm based on chaotic techniques. There are many reputed authors and papers mentioned the encryption work based on chaotic algorithm.

Efficient image encryption algorithm usually have the uniformly distributed values for cipher image, the correlation among each pixel with other pixel is very less, the key is strong enough to withstand brutal force attack, the cipher image is robust against filtering or manipulation attack, the sensitivity of the key is high and the time consume to encrypt and decrypt are less. In this proposed
work, the algorithm convinces all the above mentioned qualities of an efficient image encryption algorithm. In this paper, we compared the proposed method with the following recent papers.

Roayat Ismail Abdelfatah [5] proposed a fast double-chaotic (FDC) image encryption method. In this method initially provides the hash value of the plain image using SHA-512. Followed by that, the hash value is shifted bitwise and performed first round of permutation. In this paper, two rounds of permutations are performed. This paper also registered as good secure and efficient method than pervious methods. The speed of this algorithm is improved by choosing very less number of permutation rounds. Here, the double chaotic pseudo random generator (DCPG) and simple XOR operations are used, to increase the security in spite of reduced number of permutation operation. The DCPG requires three values of control parameters which are shared secret key. The DCPG is just the combination of both tent and Chebyshev chaotic algorithm. Also it is faster than the other previous confusion-diffusion based image encryption method.

Changzhi Yu et al. [6] introduced singular value decomposition (SVD) as an image data encryption tool. That is, here any secret image is decomposed into three different matrixes using the SVD. These decomposed segments are chaotic mapped using logistic-tent-sine system (LTSS). This method has less a very design complexity and consumes less time. It produces pixel wise less correlation ciphered image with high security. This secret image sharing scheme provides identity authentication and compression facility also.

Yanjie Songet el. [7] proposed the key-substitution architecture (KSA) scheme which is used as image encryption algorithm. This paper is mainly overcoming the time consumption and complex design problem in confusion-diffusion based image encryption algorithm. The only one round in any conventional confusion-diffusion based image encryption algorithm, the security is comparatively inadequate and the efficiency is much lesser. This KSA image encryption algorithm (KSA-IEA) has two stages in it encryption operation. The first stage is key scheming and the second stage is substitution in KSA-IEA. This key scheming method generates a ciphered image from a plain image using a chaotic algorithm based on initial key. In substitution the items of the histogram of the outcome of key scheming is scrambled to provide resistance against statistical analysis attacks [12]. The quantitative parameter chose to compare all above mentioned image encryption algorithm performance are referred and adapted from the paper proposed by Mario Preishuber et al. [8].

![Figure 1. Proposed General Secret Image Sharing Scheme](image)

2. Supporting works

2.1. Generalized Singular Value Decomposition (GSVD)
Let us assume the any matrices be A and B of size m×n and l×n respectively. All elements of A and B
belongs to real number. Then using GSVD [9], matrix A and B can be decomposed as follows:

\[ [U, W, V^T, \Phi, \Psi] = GSVD(A, B) \]

Where, \( A = UV\Phi^T \) and \( B = WV\Psi^T \)

(1)

Where, \( \Phi \) and \( \Psi \) are diagonal matrix of size \( m \times r \) and \( l \times r \) respectively.

U, W and V are orthogonal matrix of size \( m \times m \), \( l \times l \) and \( r \times r \) respectively.

To analyse GSVD [17], let us consider, \( P_a = A^T A ; P_b = B^T B \); now

\[ P = P_a + P_b. \]

(2)

Decomposing P with SVD gives, \( P = O\Omega^T \); where O is a matrix of real number of size \( n \times r \) and \( \Omega = \text{diag}(w_1, w_2, ..., w_r) \); \( w_i \geq \ldots \geq w_r > 0 \). The \( k \)-th column of matrix O is a Eigen vector of P corresponding to Eigen value \( w_k^2 \) for \( k = 1, 2, ..., r \). Then, we know,

\[ Q_A = \Omega^{-1} O^T P_a \Omega^{-1} \]

(3)

\[ Q_b = \Omega^{-1} O^T P_b \Omega^{-1} \]

(4)

\[ Q_A + Q_b = \Omega^{-1} O^T \Omega \]

(5)

Where, \( I_r \) is identity matrix of size \( r \times r \).

The spectral decomposition of \( Q_A \) and \( Q_b \) are given by,

\[ Q_A = T\Phi^2 T^T; \Phi = \text{diag}(\phi_1, \phi_2, ..., \phi_r); \]

\( \phi_i \geq 0; i = 1, 2, ..., r \)

\[ Q_b = T\Psi^2 T^T; \Psi = \text{diag}(\psi_1, \psi_2, ..., \psi_r); \]

\( \psi_i \geq 0; i = 1, 2, ..., r \)

\( \phi_i^2 + \psi_i^2 = 1; \phi_i \geq 0; \psi_i \geq 0; i = 1, 2, ..., r \)

(6)

(7)

In GSVD [9],

\[ V = O\Omega T. \]

(8)

Therefore, the matrices U and W are,

\[ U\Phi = AO\Omega^{-1} T; \text{ and } W\Psi = BO\Psi^{-1} T. \]

(9)

These equation and understanding of GSVD supports the key generation and image synthesis of both cover and secret image.

2.2. Arnold Transform

The Arnold transform is a direct confusion or scrambling method. The paper [10] explained the characteristics of this method and also, the scope for this transform to be a tool for scrambling or confusion technique. The Arnold transform is usually referred as cat mapping. The generalized Arnold transform method is,

\[
\begin{pmatrix}
  x(n+1) \\
  y(n+1)
\end{pmatrix} =
\begin{pmatrix}
  1 & a \\
  b & ab+1
\end{pmatrix}
\begin{pmatrix}
  x(n) \\
  y(n)
\end{pmatrix} \text{mod} N
\]

(10)

Where, \( a \) and \( b \) are arbitrary constant and \( N \) is size of the image or matrix which has to be scrambled. Usually, the key of Arnold transformation method is usually derived from the values \( a \), \( b \) and number
of iteration. But, in our paper we use a fixed value for a and b, only the number iterations set for scrambling plays as a key for the scrambling.

The Arnold and anti-Arnold transform used in our paper is defined as,

\[
\begin{pmatrix}
x(n+1) \\
y(n+1)
\end{pmatrix} = \begin{pmatrix}
1 & 1 \\
1 & 2
\end{pmatrix} \begin{pmatrix}
x(n) \\
y(n)
\end{pmatrix} \mod N \tag{11}
\]

And,

\[
\begin{pmatrix}
x(n+1) \\
y(n+1)
\end{pmatrix} = \begin{pmatrix}
2 & -1 \\
-1 & 1
\end{pmatrix} \begin{pmatrix}
x(n) \\
y(n)
\end{pmatrix} \mod N \tag{12}
\]

respectively. Choosing the number of iterations in Arnold transformation have limitations, that is, Arnold transform is periodic. The periodicity differs for all values of N as shown in the below table 1.

**TABLE 1. PERIODICITY OF ARNOLD TRANSFORM FOR DIFFERENT VALUE OF N**

| N | Period | N | Period |
|---|---|---|---|
| 2 | 3 | 10 | 50 |
| 3 | 4 | 12 | |
| 4 | 3 | 56 | |
| 5 | 10 | 150 | |
| 6 | 12 | | |

2.3. XOR Operation

The XOR is a very familiar Boolean operation which performs on binary inputs [11]. This XOR operation [38] is otherwise called as “complementary of equivalence”. The XOR function returns ‘true’ if values of both the input arguments of the function are different and it returns ‘false’ if values both the input argument are same. Let us assume a and b as to binary valued variables, then,

\[
(a \text{ XOR } b) \text{ XOR } b = a \tag{13}
\]

This property of the XOR function is used in the bitwise encryption of any data. Here, in this paper, XOR operator is used to encrypt the key which is described clearly in the following section. Following equation explains the operation of XOR function in encryption. Let us assume, the secret message of the same length ‘m’ to be encrypted, as ‘S(i)’ and the encryption key bit stream of length ‘m’ as ‘K(i)’, then,

\[
C(i) = [S(i)] \text{ XOR } [K(i)]; \text{ where } i=1,2,\ldots,m \tag{14}
\]

For from equation (13),

\[
S(i) = [C(i)] \text{ XOR } [K(i)]; \text{ where } i=1,2,\ldots,m \tag{15}
\]

Here, C(i) is referred to ciphered data bit stream. Equation (14) is for encryption using key ‘K’ and equation (15) is for decryption using the same key ‘K’.

3. Proposed method

3.1. Encryption process

Let us assume ‘A’ size (m*n) be the key image and ‘B’ of size (m*n) be the secret image. Also, let us assume that ‘m’ is equal to ‘n’ for all the upcoming experiments for our convenience. The proposed image encryption method includes the three significant tools for the efficient encryption of the secret image. They are 1. GSVD, 2. Arnold mapping, and 3. XOR operation. First, both the key image ‘A’ and the secret image ‘B’ are decomposed with respect to each other into two matrix of the size of image and a diagonal matrix for each image A and B separately. But, its pixel resolution of the obtained matrices may be higher than the pixels of secret image. This GSVD decomposition is actually
explained equation (1), that is key image, $A = U\varphi V^T$ and $B = W\psi V^T$. So, the encryption keys and cipher image generation procedure is illustrated in figure 2.

![Diagram](https://via.placeholder.com/150)

**Figure 2.** Generation of key and cipher image

The encryption and key generation of the proposed method is described through the figure 2. Here, the key image is shared commonly to both the encryption and decryption module. Using this key image, $A$, and secret image, $B$, are decomposed into five matrix components as discussed in equation (1). In those matrix components each element of ‘$W\psi$’ is multiplied with $10^{14}$ and rounded to nearest integer. Then, the sign bit of each element of matrix $V$ and its corresponding matrix ‘$W\psi$’ are concatenated to form a bit stream of size approximately equal to $2^{24}$ bits. This key is mentioned as ‘key_1’. The matrices $A^T A$ and $B^T B$ are obtained as shown in equation (2). By adding these two matrices, we will get a symmetric matrix ‘$P$’. The singular value decomposition (SVD) \[40\] of matrix ‘$P$’ gives, two matrices ‘$O$’ and ‘$\Omega$’. Both the matrices ‘$O$’ and ‘$\Omega$’ are multiplied with $10^{14}$ and rounded off to nearest integer. The cipher image or encrypted image, $CI$ is obtained by extracting sign bit and seven most significant bits of each element of that modified ‘$O$’. The remaining least significant bits of each element of matrix forms another key ‘key_2’. The diagonal matrix ‘$\Omega$’ is already modified into integers and now converted as a vector matrix. However, the ‘key_2’ only applied with the Arnold transform with ‘$AT_{Key}$’ rounds (AT_{Key} chose value less than N with reference to Table 1). This ‘$AT_{Key}$’ is XORed with another self-generated key, i.e., ‘$E_{Key}$’ which gives ‘$AT_{Key1}$’. Then, this both ‘$E_{Key}$’ and ‘$AT_{Key1}$’ are two keys also send to decryption module.
3.2. Decryption Process

Let us assume ‘A’ size (m*n) be the key image and ‘B’ of size (m*n). Here, we discuss the reverse process of encryption procedure in section 3.1 to decrypt and synthesis secret image. The decryption module is initially provided with ciphered image, CI, and keys (Key_1, Key_2, Key_3, E_Key and AT_Key_1). Apart from this key image, A is shared to transmitter too. By using the keys, key image, ciphered image is decrypted as illustrated in figure 3.

Initially, the keys ‘E_Key’ and ‘AT_Key_1’ is utilized to derive the ‘AT_Key’ which in turn indicates the number of iterations in Arnold transform. To rearrange the ‘Key_2’ to its actual form, i.e., the LSB of ‘O’, the inverse Arnold transform for ‘AT_Key’ times is applied. Then, through a bit level process as described in figure 3, matrix ‘O’ is formed. Using the secret key, A, a symmetric matrix $P_A$ is derived. The ‘Key_3’ is a vector matrix. Converting it into a diagonal matrix and that is $\Omega$. By using ‘O’, $P_A$ and ‘$\Omega$', a symmetric matrix $Q_A$ is derived. This $Q_A$ matrix is decomposed using SVD to get ‘T’ matrix.

As, explained in figure 3, the ‘V’ is generated by multiplying ‘O’, ‘$\Omega$’ and ‘T’. The received ‘Key_1’ is processed bitwise to separate the sign bits of ‘V’ and ‘WΨ’. The matrices ‘WΨ’ and transverse of ‘V’ is multiplied to synthesis secret image, B. this secret image, B, is synthesised without any loss as whole. This happens pixelwise in this whole procedure.

4. Simulation and Security Analysis

In this section, we present a simulation results and security comparison between other recent image encryption algorithms. Those algorithms compared with proposed method are SVD-IEA [6], FDC-IEA [5] and KSA-IEA [7]. The concern of this simulation and analyses is to evaluate the proposed algorithm with the quantitative methods. The simulation outputs, speed analysis, security against differential & noise attacks, correlation analysis, information entropy, key space and key sensitivity are the factors analysed with some quantitative techniques. Here, the image encryption scheme is performed on the grey-scale images. It scores the satisfactory numbers in the quantitative parameters (which are discussed as follows) [8].

The implementation tool configuration on which the simulation performed is a personal computer with an Intel CORE i7-6500u CPU (runs at 2.50 GHz and 8 GB RAM). The images chose here were grey-scale with resolution 512*512. The initial conditions applied for the proposed GSVD based image
encryption algorithm are the Ex-OR key, E\_Key of value less than 255, the Arnold transform key, AT\_Key of value less than the value mentioned in table 1 (and it depends on the size of secret image) and the key image, A is chose with different less significant images.

4.1. Simulation Results

Here, a set of 10 images are used in the simulation process for different set of experiments and result analysis as mentioned in the table 2. For the proposed image encryption scheme general simulation we use four familiar images (grayscale). They are images of lena, cameraman, baboon and airplane. All these images have resolution of about 512*512. The chosen key image is in tagged image file format (tiff or tif). The simulation results show a satisfactory performance on resisting statistical attack. However, the decrypted image obtained from the ciphered image using proposed scheme is lossless.

The simulation result images are displayed in the fig. 4, fig. 5, fig. 6, fig. 7 and fig. 8. In the fig. 4, four original images are displayed, Lena, Cameraman, Baboon and Airplane in (a), (b), (c) and (d) respectively. In fig. 5, the ciphered images using proposed encryption scheme on the corresponding original images in fig. 4. The images decrypted from the ciphered images in fig. 5 are displayed in fig. 6 respectively. In fig. 7 (a,b,c,d), the histograms of the original images in fig. 4(a,b,c,d) are displayed respectively. In fig. 8, the histograms of the corresponding ciphered image shown in fig. 5 are displayed. So, the performance of the encryption method is satisfactory for various images. The simulation results find that the histogram of the ciphered image is uniformly distributed which shows the randomness of the values of the encrypted image.

4.2. Speed Analysis

The speed analysis refers the execution time of the image encryption scheme. This execution time consumption is an important factor for any image encryption algorithm. Here, we compared our proposed method with few recent works. They are SVD-IEA [6], FDC-IEA [5] and KSA-IEA [7]. Here, to analyse the speed of the image encryption algorithm, we calculated the time consumed by the algorithm to encrypt the colour and grayscale image of size 512*512.

| No. | Image      | SVD-IEA [6] | FDC-IEA [5] | KSA-IEA [7] | RGSVD-IEA |
|-----|------------|-------------|-------------|-------------|-----------|
| 1   | Airplane   | 0.8111      | 0.5479      | 0.3556      | 0.3334    |
| 2   | Baboon     | 0.7684      | 0.6074      | 0.3360      | 0.3252    |
| 3   | Barbara    | 0.7767      | 0.5532      | 0.3379      | 0.3229    |
| 4   | Bridge     | 0.8031      | 0.5701      | 0.3336      | 0.3299    |
| 5   | Clown      | 0.7622      | 0.5447      | 0.3329      | 0.3282    |
| 6   | Couple     | 0.7683      | 0.5454      | 0.3626      | 0.3498    |
| 7   | Elain      | 0.8082      | 0.5089      | 0.3288      | 0.3178    |
| 8   | Frog       | 0.7714      | 0.5084      | 0.3367      | 0.3321    |
| 9   | Girlface   | 0.7999      | 0.5721      | 0.3362      | 0.3311    |
| 10  | Goldhill   | 0.7843      | 0.5259      | 0.3403      | 0.3343    |
For the simulation of the scheme, we used Matlab 2019a software in a computer with the same specification as mentioned in beginning this section. The results are displayed in the table 2. This comparison provides enough evident that the proposed scheme have a satisfactory speed on most cases.

4.3. Differential Attack

The image encryption scheme is robust against the differential attack, only if even the small change of value in the secret image produces totally different ciphered image. This property can be analyzed quantitatively using two parameters, they are, the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). These parameters evaluate the ability of any encryption scheme to resist the differential attack. The NPCR and UACI is defined as follows in equation (16).
**Table 3. NPCR results of five image encryption algorithms for grayscale images**

| No. | Image  | SVD-IEA [6] | FDC-IEA [5] | KSA-IEA [7] | RGSVD-IEA |
|-----|--------|-------------|-------------|-------------|-----------|
| 1   | Airplane | 0.043255    | 0.995948    | 0.996038    | 0.996176  |
| 2   | Baboon  | 0.043255    | 0.996041    | 0.996017    | 0.996137  |
| 3   | Barbara | 0.043255    | 0.996173    | 0.996049    | 0.996104  |
| 4   | Bridge  | 0.043255    | 0.995806    | 0.996162    | 0.996194  |
| 5   | Clown   | 0.043255    | 0.996131    | 0.996051    | 0.996278  |
| 6   | Couple  | 0.043255    | 0.996301    | 0.996188    | 0.996239  |
| 7   | Elain   | 0.043255    | 0.995928    | 0.995874    | 0.995990  |
| 8   | Frog    | 0.043255    | 0.995989    | 0.996169    | 0.996231  |
| 9   | Girlface| 0.043255    | 0.996071    | 0.996213    | 0.996243  |
| 10  | Goldhill| 0.043255    | 0.995929    | 0.995918    | 0.996065  |

**Figure 7.** Histogram of original Secret Image (a) Lena; (b) Cameraman; (c) Baboon; (d) Airplane

**Figure 8.** Histogram of Ciphered Image (a) Lena; (b) Cameraman; (c) Baboon; (d) Airplane

**Table 4. UACI results of five image encryption algorithms for grayscale images**

| No. | Image  | SVD-IEA [6] | FDC-IEA [5] | KSA-IEA [7] | RGSVD-IEA |
|-----|--------|-------------|-------------|-------------|-----------|
| 1   | Airplane | 0.000170    | 0.334324    | 0.335248    | 0.335698  |
| 2   | Baboon  | 0.000170    | 0.334111    | 0.334349    | 0.334894  |
| 3   | Barbara | 0.000170    | 0.334794    | 0.334649    | 0.335246  |
| 4   | Bridge  | 0.000170    | 0.334634    | 0.334748    | 0.334998  |
| 5   | Clown   | 0.000170    | 0.334205    | 0.334683    | 0.334811  |
| 6   | Couple  | 0.000170    | 0.334834    | 0.334037    | 0.334443  |
| 7   | Elain   | 0.000170    | 0.334275    | 0.334053    | 0.334653  |
| 8   | Frog    | 0.000170    | 0.334492    | 0.334374    | 0.334787  |
| 9   | Girlface| 0.000170    | 0.334549    | 0.334273    | 0.335318  |
| 10  | Goldhill| 0.000170    | 0.334965    | 0.334729    | 0.334826  |
Where the $C_1(i,j)$ and $C_2(i,j)$ are the value of pixel in the position $(i,j)$ of two different ciphered image. The $C_1$ is the ciphered image of actual secret image and $C_2$ is the ciphered image of slightly changed secret image. The $M*N$ denotes the resolution of the image. Then, the $D$ is a matrix depends on $C_1$ and $C_2$, that is if $C_1(i,j) \neq C_2(i,j)$, then $D(i,j)=1$ and otherwise $D(i,j)=0$. In table 3 and table 4, the observed values of the NPCR and UACI of the proposed scheme are compared with the four recent methods. Usually, to resist chosen /known plaintext attack, any satisfactory image encryption algorithm should be sensitive to small changes in the plain-image. From the observation, the above mentioned data clearly depicts that the proposed image encryption scheme is very sensitive to the change in plain text.

4.4. Correlation Analysis

The correlation analysis is generally observed with very high value in original secret image. Here, the correlation analysis actually means the pixel wise similarity of any image in horizontal, vertical and diagonal direction. It is clearly defined in equation (17).

$$
\rho = \frac{E[(x - E(x))(y - E(y))]}{D(x)D(y)}
$$

$$
E(x) = \frac{1}{\omega} \sum_{i=1}^{\omega} x_i
$$

$$
D(x) = \frac{1}{\omega^2} \sum_{i=1}^{\omega} [x_i - E(x)]^2
$$

(17)

Where the operator $E[.]$ is expectation and $D[.]$ is variance. The variable $x$ and $y$ represents the actual set of values of the ciphered image and the shifted (in horizontal, vertical and diagonal directions) set of values of the ciphered image. Any good image encryption scheme should produce a ciphered image with less correlation coefficient ($\rho$). The correlation coefficient always varies between the values $[0,1]$. In fig. 9 and fig. 10, we present the scatter plot of the secret images (Lena) for the horizontal, vertical and diagonal direction correlation coefficient values.

In table 5, the observed values of the correlation coefficient of the plain image and ciphered image encrypted using proposed scheme are compared with the four other recent schemes for grayscale image (Lena) of size 512*512.
TABLE 5. CORRELATION COEFFICIENTS OF FIVE IMAGE ENCRYPTION ALGORITHMS FOR GRAYSCALE IMAGE (LENA 512*512)

| Direction | Plain-image | SVD-IEA [6] | FDC-IEA [5] | KSA-IEA [7] | RGSVD-IEA |
|-----------|-------------|-------------|-------------|-------------|-----------|
| Horizontal | 0.977898    | 0.001973    | 0.001869    | 0.001844    | 0.001341  |
| Vertical   | 0.992487    | 0.000170    | 0.334111    | 0.334439    | 0.334894  |
| Diagonal   | 0.973249    | 0.000170    | 0.334794    | 0.334649    | 0.334546  |

4.5. Key Space and Key Sensitivity

For a good image encryption scheme, the key space size should be at least $2^{100}$. This is a sufficient value to resist brute-force attack. In our proposed scheme, the key space size is certainly greater than $2^{100}$.

To check the sensitivity of the key of the proposed scheme, a slight change in the value of key is applied to decrypt the image. The result of this simulation is shown in fig. 10 (b). The decrypted image does not contain any information of the secret image. Here, we chose a grayscale image for analysis. That image is Lena.tif and its resolution is 512*512. The decrypted image with the slightly changed key is totally degraded and there is no influence of the secret image.

4.6. Noise Attack

Any good image encryption algorithm should be very robust against the noise attack. To check the robustness of the proposed scheme against the noise attack, we add the Gaussian noise (with mean as
‘0’ and variance as 0.1) to the cipher-image. Then, the simulation decrypts that “noise added” cipher-image. This simulation results are present in the fig. 10 (c), which proves the ability of the proposed scheme against noise attack. In fig. 10 (d), the simulation results of the effect of ‘salt and pepper’ noise on the ciphered image before decryption are displayed. The ciphered image is corrupted with the noise density of 0.2. Then, synthesized output image is display in fig. 10 (d). The performance of the scheme is good on both Gaussian and ‘salt & pepper’ noises.

5. Conclusion
This work proposes a novel image encryption scheme based on the regional generalized singular value decomposition (GSVD). This method is compared with FDC-IEA, SVD-IEA and KSA-IEA. From the simulation results and discussion on that topic, the performance of the proposed scheme is satisfactory. The proposed scheme avoids the complexities in the conventional confusion-diffusion based methods. So, from the result analyses and comparison tables, the speed of the encryption method, the robustness of the scheme against differential attack (by NPCR and UACI), robustness against noise attack, key space and key sensitivity are performing better than the other methods. The correlation results shows the reliable randomness among the bits of encrypted image. This pixelwise GSVD based image encryption method may provide space for further enhancements in future. The hardware implementation of this algorithm is feasible and is planned is future.

References
[1] Issam H, Kamal S, Ezz M 2010 High-Speed AES Encryptor With Efficient Merging Techniques IEEE Embedded Systems Letters 2
[2] Rijmen V 2004 Equivalent descriptions for the DES Electronics Letters 40
[3] Meghrajani Y K, Desai L S and Mazumdar H S 2019 Secure and efficient arithmetic-based multi-secret image sharing scheme using universal share Journal of Information Security Application 47 267–74
[4] Zhu H, Zhao Y and Song Y 2019 2D Logistic-Modulated-Sine-Coupling-Logistic Chaotic Map for Image Encryption IEEE Access 7 14081–98
[5] R I Abdelfatah 2019 A new fast double-chaotic based Image encryption scheme Multimed. Tools Applications
[6] P Singh, B Raman and M Misra 2017 A secure image sharing scheme based on SVD and Fractional Fourier Transform Signal Processig Image Communication 57 46–59
[7] Y Song, Z Zhu, W Zhang, H Yu and Y Zhao 2019 Efficient and Secure Image Encryption Algorithm Using a Novel Key-Substitution Architecture IEEE Access 7 84386–400
[8] M Preishuber, T Hutter, S Katzenbeisser and A Uhl 2018 Depreciating motivation and empirical security analysis of chaos-based image and video encryption IEEE Transaction Information Forensics Security 13 2137–50
[9] S Friedland 2006 A new approach to generalized singular value decomposition SIAM Journal Matrix Analytics Application 27 434–44
[10] F Chen, K Wong, X Liao and T Xiang 2014 Period distribution of generalized discrete Arnold cat map Theoretical Computational Science 552 13–25
[11] Y Qin and Y Zhang 2017 Information Encryption in Ghost Imaging With Customized Data Container and XOR Operation IEEE Photonics Journal 9 1–8
[12] R Ge, G Yang, J Wu, Y Chen, G Coatrieux and L Luo 2019 A Novel Chaos-Based Symmetric Image Encryption Using Bit-Pair Level Process IEEE Access 7 99470–80