Reversible De-identification of Specific Regions in Biomedical Images and Secured Storage by Randomized Joint Encryption

Prabhavathi K[1], Anandaraju M. B[2]
Research Scholar, Department of ECE, BGSIT, Adichunchanagiri University, B. G. Nagara, India[1]
Professor and HOD, Department of ECE, BGSIT, Adichunchanagiri University, B. G. Nagara, India[2]

***Abstract*—In many circumstances, de-identification of a specific region of a biomedical image is necessary. De-identification is used to hide the subject's identity or to prevent the display of the objectionable or offensive region(s) of the image. The concerned region can be blurred (de-identified) by using a suitable image processing technique guided by the region-defining mask. The proposed method provides lossless blurring, which means the original image can be recovered fully with zero loss. The blurred image and the region-defining mask, along with the digital signature, are jointly encrypted to form the composite cipher matrix, and it is stored in the cloud for further distribution. The composite cipher matrix is decrypted to recover the blurred image by the conventional end user. Further, using the deblur key, the original image can be recovered with zero loss by the fully authorized special end users. On decryption, the digital signature is available for both types of end users. The proposed method uses randomized joint encryption using integer matrix keys in a finite field. The experimental results show that the proposed method achieves a reduction in the average execution time of encryption by 30 to 40 percent compared to its nearest competitor. Additionally, the proposed scheme achieves very nearly ideal performance with reference to the correlation coefficient, entropy, pixel change rate, and structural similarity index. Overall, the proposed algorithm performs substantially better than the other similar existing schemes for large-sized images.**

Keywords*—Region identification mask; modular matrix inverse; selective image encryption; image de-identification; randomized joint encryption; image authentication

I. INTRODUCTION

When intimate and informative images, like medical, forensic, personal, etc., are stored in the cloud, it is essential to provide privacy and security to those images [1]. This can be achieved using steganography or encryption [2]. Each method has its own advantages and limitations. In this work, the image encryption route is chosen where image and source authentications are implemented concurrently with encryption. The image encryption process can be full or selective. In full image encryption [3-7], the entire image is encrypted, while in Selective Encryption (SE), only certain specific parts of the image are encrypted.

On many occasions, explicitly specified regions of the given image are selected and obscured (blurred) due to personal privacy, societal or legal requirements, censorship guidelines, or to hide embedded textual information and so on [8].

The non-selected region is retained without any change to convey the desired visual information. In the given image, the specific region to be de-identified (blurred or obscured) is referred to as the Region of Interest (ROI). The ROI gives the location of the image objects, like face, iris, personal textual data, and parts to be censored, *etc*. ROI locations are obtained by image segmentation procedures, as explained in [9]. In general, ROI regions are represented by the Region Defining Binary Mask (RD-BM) where the pixels of the ROI are set to 1's and the other pixels to 0's. An ROI can be marked manually also by the visual inspection of the image. Once the ROI is determined, the de-identification of that region is carried out by Selective Encryption (SE) [10-14].

In many cases, specially authorized users (like investigating agencies, medical image diagnostic units, *etc*.) should be able to recover the original unobscured image [15-19]. In these cases, exact reverse de-identification is required, and it is skillfully implemented in the proposed method. Here, XOR encoding is employed for selective encryption while its vulnerability to chosen Plaintext Attack (C-PA) is eliminated by randomizing the encryption key for successive encryptions. Thus, every encryption process uses a different encryption key so that the present key, if captured by an attacker, is no longer valid for the next encryption.

Image authentication ensures the integrity of the encrypted images [20-27]. In the proposed method, a hidden encrypted matrix acts as the digital signature that provides authentication for the encrypted primary image. The signature matrix is decoded and verified by the end user, and if the verification fails, the image decryption process is terminated.

Digital images are represented by matrices whose elements belong to the data type uint8. Therefore, in this work, integer matrix keys and matrix operations in a specific finite field Zp, are used for the encryption and decryption of image matrices to achieve faster cryptographic operations. By using the finite field, all cryptographic operations are carried out in the integer domain, and thus, the round of error that occurs with floating point operations is eliminated. Finite field arithmetic also limits the maximum value of any element to (p–1), and thus integer overflow problem is avoided.

The objectives of this work are:

1) Fully reversible de-identification of the selected regions of the given image.

2) Security for the de-identified image during transmission and storage.

3) Easy decryption of the encrypted image to get back the de-identified image by the conventional end users.

4) Lossless recovery of the original image by specially authorized users.

5) Content and source authentication via a digital signature scheme.

In achieving these objectives, the target image, the region identification mask, the signature matrix, and the randomization matrix are jointly encrypted to get a composite cipher matrix. Thus the proposed method is designated as Reversible De-identification of Specific Regions by Randomized Joint Encryption, RDSR-RJE.

The rest of this paper is organized as follows. Section II contains the literature review. Section III introduces the mathematical operations used for the generation of cryptographic matrix keys. Section IV describes the encryption and decryption techniques of RDSR-RJE. Section V holds the experimental results and the performance evaluation. Section VI gives the conclusion.

II. Literature Review

Plenty of research articles are available on image encryption using diverse methods. Kaur and Kumar [3], have presented an extensive survey of various image encryption schemes and the corresponding algorithms. Several commonly used metrics, namely key space analysis, image entropy, correlation coefficients, MSE, PSNR, SSIM, sensitivity analysis, and execution time, are discussed in detail. Sajitha and Rekh [4], have reviewed recent image encryption methods, including a broad coverage of the relative advantages and disadvantages of different methods. Chaos-based image encryption schemes are reviewed in [5]. Various types of chaotic maps used in image encryption are enumerated systematically. In [6], the authors have reviewed several image encryption and discrete image encoding techniques and the associated future scope in those fields. In [7], image encryption methods based on chaotic maps, neural networks, AES (Advanced Encryption Standard), DCT (direct Cosine Transform), XOR, GA (Genetic Algorithm), and LSB (Least Significant Bit) are reviewed. Additionally, several schemes which use a combination of these methods are discussed. In [8], the author has differentiated between anonymization and de-identification, where the former procedure is irreversible, and the latter is reversible.

A few survey papers are available on SE with Reversible De-identification (RD) [10-12]. Selective Color image encryption with RD is presented in [10]. In [11], the authors have reviewed different techniques for the SE of multi-media content. In [12], applications of SE in the Covid-19 environment have been presented. In [13], the authors use permutation method for confusion and XOR for diffusion. However, the block wise approach increases the time complexity of encryption as well as decryption. In [14], dynamic DNA coding along with a sine function based chaotic map is used for encryption. However, the security level offered by DNA coding is moderate, while the calculations involved in DNA coding result in higher computational overhead. Additionally, the sine map has a relatively smaller chaotic interval and a low-security level. In [15], threshold entropy and the Arnold Cat Map are used for lossless selective encryption. But, the block-wise calculation of entropy incurs a higher computational cost, especially for large-sized images. In [16], the authors have adopted multi-level encryption with compressive sensing and have used degradation matrices to implement reversible, selective encryption. However, the reverse recovery of the obfuscated regions is not error-free due to the least square estimation. In [17], The SE is carried out by the permutation of pixels block-wise on the selected region, and then the de-identified image can be encoded further by JPEG-like compressive encoding. The authors have extended this operation for video surveillance. But the block-wise operation is computationally quite expensive, and moreover, the ROI selected has to be a square block which is another limitation. In [18], the authors have used the Histogram Shifting method to hide the ROI in a high-textured part of the image. Then, the Arnold map technique is used to obfuscate the ROI. After this, the QR code is used for the overall encryption. The major deficiency of this method is the trial and error procedure used to select the high-textured area. Additionally, this area gets distorted after decryption due to the data-hiding mechanism. Therefore, even though the ROI is recovered, the full image recovered is a slightly distorted version of the original one. In [19], the authors have used the 'reversible data hiding (RDH)' technique so that the obfuscated ROI can be recovered back at the receiving side. RDH is achieved using 'difference value embedding', which is a block-wise approach. Additionally, Run Length Encoding (RLC) and Huffman coding are used to improve the hiding capacity. These additional encodings and the block-wise difference operations make this method computationally very expensive, even for a relatively small sized image.

Now, a few existing works on encryption with authentication are briefly discussed. In [20], the authors have used AES for the encryption of the main image and ECC for hiding the AES encryption key and the hash values corresponding to the image and source authentication. The Dicom file header is chosen as the hiding location. Thus the security of the authentication signatures is not really strong. In [21], Elliptic Curve Cryptography (ECC) along with 3D/4D Cat mappings, are used for image encryption. Image authentication is provided using SHA_{256}. However, the block-wise operations and the use of ECC having a 512-bit prime order make the computational cost very high. In [22], compressive sensing and the Logistic-Tent system are used for image encryption along with blind signcryption based on secret sharing and ECC. Here, the additional DWT transform at encryption, and IDWT at decryption increases the time complexity excessively. In [23], the authors have used ECC for digital signature and the Logistic Tent map for image encryption. Use of chaotic maps for both permutation and diffusion results in higher computational complexity. In [24], two-stage encryption has been adopted. The first stage uses

ECC for asymmetric encryption, and the second stage implements multi-chaotic maps for confusion. The authors have used SHA_{256} for authentication. Here, the group formation and the generation of big integers for ECC, introduce an inordinate level of computational complexity for large-sized images. Additionally, the transmission of the digital signature part separately along with the encrypted image, increases the vulnerability for a security breach. In [25], the authors have used Equal Absolute Value Decomposition preceded by Fresnal transform to achieve optical image encryption. Image authentication is implemented based on nonlinear correlation. But the disadvantage of this method is that its security can be compromised using the amplitude phase-retrieval algorithm. In [26], differential privacy (DP) schemes, for selective image encryption, based on different techniques have been evaluated. The authors have concluded that the method using Singular Value Decomposition (SVD) provides the best solution. However, the DP methods discussed in this review use block-wise processing and thus suffer inordinate time delay during encryption. In [27], 2D logistic sine-cosine maps are used for confusion, and Mandelbrot Set and conditional shift algorithms, along with XOR, are used for diffusion. However, the conditional shift algorithm used introduces a substantial time delay. In [28], the authors have used a 3D chaotic map for position permutation (confusion) as well as value transformation (diffusion) that involves pixel rotation. However, the histogram equalization process and the generation of a third-order chaotic map consume quite a long execution time both during encryption and decryption. In [29], the spatial, as well as the frequency domain approach, has been used for image encryption. Block scrambling based on a randomizing key provides confusion, and the 2D Logistic Sine Map (LSM) coupled with wavelet transform coefficients provide diffusion. SHA_{512} hash of the image is used as the input to drive the LSM. However, the whole process is highly complex, and the execution speed is very low. In [30], the authors have used the Mandelbrot set, DNA sequence technique, and a suitable chaotic map for the encryption of color images. The chaotic map to be used is evaluated and selected based on the entropy of the image under consideration. However, the entire operation of encryption/decryption incurs heavy computational overhead due to repeated calculations of entropy and generation of the appropriate Mandelbrot set for each image. In [31], a cosine transform-based chaotic system (CTBCS) has been employed for image encryption. With two seed maps, CTBCS acquires complex dynamic characteristics that provide a higher degree of security. From CTBCS, the authors have derived a logistic sine cosine map, sine Tent cosine map, and Tent logistic cosine map. The encryption is carried out in multi-stages using these maps. Here, even though the security level achieved is very high, the encryption process is block-wise and extensive, which imposes an inordinate computational burden for large-sized images. In [32], the authors have adopted matrix semi-tensor product (STP) for image diffusion. Additionally, Boolean network-based compound secret keys are used for confusion. In this case also, the block-wise approach introduces quite a time delay for the encryption/decryption of images. In [33-37], image encryption schemes based on machine learning have been presented, and the superiority of training-based deep learning networks has been established. Since these methods belong to an entirely different genre, they are not discussed in this work.

Most of the schemes discussed here have a high degree of time complexity in the key generation as well as encryption/decryption processes due to the block-wise operations and iterative algorithms using floating point data types. Thus the execution times are higher for large-sized images. In our proposed method RDSR-RJE, the time complexity is less as there are no block-wise or iterative operations. In RDSR-RJE, all operations are carried out in the integer domain that achieves higher speed.

III. PRELIMINARIES

In RDSR-RJE, the key spaces as well as calculations involving encryption, are carried out using modular algebra in the finite field Zp where its members are integers in the range 0 to (p–1).

A. Basic Modular Operations Extended to Matrices

The basic modulo operation is represented in a few ways as $b = a \bmod p$; $b = a\%p$; $b = mod(a, p)$. In this paper, we use the notation $b = mod(a, p)$, where $mod(\ldots)$ acts as a function that returns the modulo remainder. The $mod(\ldots)$ function can be easily extended to integer matrices in Zp. Let $\boldsymbol{A}$ be an integer matrix of size $(m \times n)$ whose elements are $a(i, j)$'s for $i = 1$ to m and $j = 1$ to n. Now, the $mod(\ldots)$ function is extended to matrix A, simply as $mod(\boldsymbol{A}, p)$. Here, $mod(\ldots)$ function is applied element-wise to all $a(i, j)$'s. To clarify, let matrix $\boldsymbol{B}$ represent the result of $mod(\boldsymbol{A}, p)$ as,

$$\boldsymbol{B} = mod(\boldsymbol{A}, p) \qquad (1)$$

Then,

$$b(i,j) = mod(a(i,j), p) \qquad (2)$$

for $i = 1$ to m and $j = 1$ to n. The elements of $\boldsymbol{B}$ belong to Zp and $\boldsymbol{B} \in Z_p^{m \times n}$. The basic scalar identities of $mod(\ldots)$ functions, as well as the associative and distributive laws, hold good for modular matrix operations.

B. Modular Matrix Inverse

Modular matrix inverse of a square matrix $\boldsymbol{A}$ of size $n \times n$, represented by $mmi(\boldsymbol{A}, p)$, is defined such that,

$$\text{mod}(\mathbf{A} * \text{mmi}(\mathbf{A}, \text{p}), \text{p}) = \text{mod}(\text{mmi}(\mathbf{A}, \text{p}) * \mathbf{A}, \text{p}) = \mathbf{I}_{\text{n}\times\text{n}}$$

For the existence of $mmi(\boldsymbol{A}, p)$ the rank of $\boldsymbol{A}$ should be n, and p should be prime, which assures that the $GCD(det(\boldsymbol{A}), p) = 1$.

Rectangular integer matrices have either left $mmi(\ldots)$'s or right $mmi(\ldots)$'s. A rectangular matrix $\boldsymbol{E}$ of size $m \times n$ with $m > n$ (tall matrix) and rank n, has the left $mmi(\ldots)$ only, which means,

$$\text{mod}(\text{mmi}(\mathbf{E}, \text{p}) * \mathbf{E}, \text{p}) = \mathbf{I}_{\text{n}\times\text{n}}$$

When $n > m$, (wide matrix), $\boldsymbol{E}$ having rank m, has the right $mmi(\ldots)$ as,

$$\text{mod}(\mathbf{E} * \text{mmi}(\mathbf{E}, \text{p}), \text{p}) = \mathbf{I}_{\text{m}\times\text{m}}$$

Detailed determination of $mmi(\boldsymbol{E}, p)$ for a given matrix $\boldsymbol{E}$ and p, is described in the next section.

C. Generation of Encryption and Decryption Matrix Keys

In RDSR-RJE, encryption and decryption operations use four distinct integer matrix keys of size (n×n) each. An efficient generation of these keys is based on the modified Householder Construction [38].

1) Modified householder construction: Conventional Householder Construction (CHC) generates an orthogonal symmetric matrix from a given vector. The basic CHC equation [40] is,

$$\boldsymbol{H} = \boldsymbol{I}_{L\times L} - \frac{2*V*V^T}{V^T*V} \quad (3)$$

where V is a column vector of size L×1 and $\boldsymbol{I}_{L\times L}$ is the identity matrix. It can be verified that $\boldsymbol{H}$ is an L×L orthogonal matrix, and it is symmetric where all the elements are not independent.

In cryptography, with a symmetric secret matrix key, the number of unknown elements in a matrix gets reduced almost by 50%, which makes the brute force guessing task easy. Therefore, for better security, secret keys should neither be symmetric nor based on symmetric parent matrices. To get an unsymmetric involutory matrix in Zp, the CHC procedure is modified using two dissimilar random integer vectors U and V of size L×1 to get $\boldsymbol{G}$ as,

$$\boldsymbol{G} = \boldsymbol{I}_{L\times L} - \frac{2*U*V^T}{U^T*V} \quad (4)$$

The RDSR-RJE scheme uses modular algebra, and the keys derived from G must be integers. Therefore, $\boldsymbol{G}$ has to be an integer matrix in the finite field Zp. To get this, the division operation by $(U^T * V)$ in (4) is replaced by the multiplication factor $(U^T * V)^{-1}$ which is the modular inverse of $(U^T * V)$ with respect to p. The resulting $\boldsymbol{G}$ is,

$$\boldsymbol{G} = I_{L\times L} - 2 * (U^T * V)^{-1} * U * V^T \quad (5)$$

In (5), the size of $\boldsymbol{G}$ is L×L, and all the mathematical operations are carried out using modular algebra in the finite field Zp. Here, it can be verified that,

$$mod(\boldsymbol{G} * \boldsymbol{G}, p) = \boldsymbol{I}_{L\times L} \quad (6)$$

That is, the modular inverse of $\boldsymbol{G}$ is $\boldsymbol{G}$ itself. That means $\boldsymbol{G}$ is involutory and not well suited as a cryptographic key from the security aspect. Hence, to avoid the involutory deficiency, an additional involutory matrix $\boldsymbol{F}$, which is entirely different from $\boldsymbol{G}$, is generated as,

$$\mathbf{F} = I_{L\times L} - 2 * (X^T * Y)^{-1} * X * Y^T \quad (7)$$

Here X and Y are two L×1 integer vectors different from U and V. Similar to matrix $\boldsymbol{G}$, we have,

$$mod(\boldsymbol{F} * \boldsymbol{F}, p) = \boldsymbol{I}_{L\times L} \quad (8)$$

Let us define two integer matrices $\boldsymbol{E}$ and $\boldsymbol{D}$ as,

$$\boldsymbol{E} = mod(\boldsymbol{G} * \boldsymbol{F}, p) \quad (9)$$

$$\boldsymbol{D} = mod(\boldsymbol{F} * \boldsymbol{G}, p) \quad (10)$$

Since $\boldsymbol{G}$ and $\boldsymbol{F}$ are derived from different vector sets, they are non-commutative. That is, $(\boldsymbol{G} * \boldsymbol{F}) \neq \boldsymbol{F} * \boldsymbol{G}$.Therefore, the encryption and decryption parent matrices $\boldsymbol{E}$, and $\boldsymbol{D}$ are numerically dissimilar.

Now, let us evaluate the products $\boldsymbol{E}*\boldsymbol{D}$ and $\boldsymbol{D}*\boldsymbol{E}$. For easier writing, the mod prefix and p are omitted while writing the expressions for the matrices. Then,

$$\boldsymbol{Z} = mod(\boldsymbol{E} * \boldsymbol{D}, p) = \boldsymbol{E} * \boldsymbol{D} \quad (11)$$

$$\boldsymbol{W} = mod(\boldsymbol{D} * \boldsymbol{E}, p) = \boldsymbol{D} * \boldsymbol{E} \quad (12)$$

From (9), (10), and (11),

$$\boldsymbol{Z} = (\boldsymbol{G} * \boldsymbol{F}) * \boldsymbol{F} * \boldsymbol{G} = \boldsymbol{G} * (\boldsymbol{F} * \boldsymbol{F}) * \boldsymbol{G} \quad (13)$$

Substituting (8) and then (6) in (13) gives,

$$\boldsymbol{Z} = \boldsymbol{E} * \boldsymbol{D} = \boldsymbol{I}_{L\times L} \quad (14)$$

Similarly, it can be shown that,

$$\boldsymbol{W} = \boldsymbol{D} * \boldsymbol{E} = \boldsymbol{I}_{L\times L} \quad (15)$$

From (14) and (15), it can be seen that $\boldsymbol{D}$ is the modular matrix multiplicative inverse of $\boldsymbol{E}$ and vice versa as, $\boldsymbol{D}$ = *mmi*($\boldsymbol{E}$, p) and $\boldsymbol{E}$ = *mmi*($\boldsymbol{D}$, p). In (14) and (15), $\boldsymbol{E}$ and $\boldsymbol{D} \in Z_p^{L\times L}$. The novelty of generating *mmi*'s by the Householder technique is that mmi's are obtained without directly calculating the matrix inverses in Zp (with time complexity $O(L^3 * log_2 p)$ [39]), but using only scalar modular inverse as in (5) and (7), which has a time complexity of $O((log_2 p)^2)$ only [40].

2) Encryption and decryption matrix keys from matrices $\boldsymbol{E}$ and $\boldsymbol{D}$: Individual encryption keys are obtained by the row-wise splitting of the parent matrix **E** of size L×L, into four sub matrices $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, and $\mathbf{E}_4$ as,

$$\boldsymbol{E}_{L\times L} = \begin{bmatrix} \boldsymbol{E}_{1,(n\times L)} \\ \boldsymbol{E}_{2,(n\times L)} \\ \boldsymbol{E}_{3,(2\times L)} \\ \boldsymbol{E}_{4,(2\times L)} \end{bmatrix} \quad (16)$$

In (16), the selected size of each sub matrix is marked in its subscript.

3) Selection of the sizes of $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ and $\boldsymbol{E}_4$: In RDSR-RJE, the matrix keys $\mathbf{E}_1$ and $\mathbf{E}_2$ are used to encrypt the plain image matrices of size k×n by post multiplication. Therefore, the number of rows of $\mathbf{E}_1$ and $\mathbf{E}_2$ are set to n to match the column size of the plain image matrices. On the other hand, $\mathbf{E}_3$ and $\mathbf{E}_4$ are used to encrypt the signature matrix and the randomizing matrix, whose sizes are k×2. Hence the row sizes of $\mathbf{E}_3$ and $\mathbf{E}_4$ are set to 2. The detailed encryption process is given in section IV, where the significance of the sizes of submatrices $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, and $\mathbf{E}_4$ will become clear. The choice of using 2 instead of n, for the row sizes of $\mathbf{E}_3$ and $\mathbf{E}_4$, is to reduce the overall key sizes and the resulting cipher matrix size to keep the ciphertext expansion ratio at a lower value.

The total number of rows of the sub matrices $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$, $\boldsymbol{E}_4$ is (n+n+2+2) = 2*n+4. Therefore from (16), it can be seen that

$$L = 2*n + 4 \quad (17)$$

The decryption keys are obtained by the column-wise splitting of the parent matrix $\boldsymbol{D}$ into four submatrices as,

$$\boldsymbol{D}_{L\times L} = [\boldsymbol{D}_{1,(L\times n)} \quad \boldsymbol{D}_{2,(L\times n)} \quad \boldsymbol{D}_{3,(L\times 2)} \quad \boldsymbol{D}_{4,(L\times 2)}] \quad (18)$$

Now substituting in (14), for $\boldsymbol{E}$ and $\boldsymbol{D}$ from (16) and (18) gives,

$$\begin{bmatrix} \boldsymbol{E}_1 \\ \boldsymbol{E}_2 \\ \boldsymbol{E}_3 \\ \boldsymbol{E}_4 \end{bmatrix} * [\boldsymbol{D}_1 \quad \boldsymbol{D}_2 \quad \boldsymbol{D}_3 \quad \boldsymbol{D}_4] = \boldsymbol{I}_{L\times L} \quad (19)$$

Now, expanding the LHS and the RHS of (19) in terms of compatible submatrices gives,

$$\begin{bmatrix} \boldsymbol{E}_1 * \boldsymbol{D}_1 & \boldsymbol{E}_1 * \boldsymbol{D}_2 & \boldsymbol{E}_1 * \boldsymbol{D}_3 & \boldsymbol{E}_1 * \boldsymbol{D}_4 \\ \boldsymbol{E}_2 * \boldsymbol{D}_1 & \boldsymbol{E}_2 * \boldsymbol{D}_2 & \boldsymbol{E}_2 * \boldsymbol{D}_3 & \boldsymbol{E}_2 * \boldsymbol{D}_4 \\ \boldsymbol{E}_3 * \boldsymbol{D}_1 & \boldsymbol{E}_3 * \boldsymbol{D}_2 & \boldsymbol{E}_3 * \boldsymbol{D}_3 & \boldsymbol{E}_3 * \boldsymbol{D}_4 \\ \boldsymbol{E}_4 * \boldsymbol{D}_1 & \boldsymbol{E}_4 * \boldsymbol{D}_2 & \boldsymbol{E}_4 * \boldsymbol{D}_3 & \boldsymbol{E}_4 * \boldsymbol{D}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times n} & \boldsymbol{0}_{n\times 2} & \boldsymbol{0}_{n\times 2} \\ \boldsymbol{0}_{n\times n} & \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times 2} & \boldsymbol{0}_{n\times 2} \\ \boldsymbol{0}_{2\times n} & \boldsymbol{0}_{2\times n} & \boldsymbol{I}_{2\times 2} & \boldsymbol{0}_{2\times 2} \\ \boldsymbol{0}_{2\times n} & \boldsymbol{0}_{2\times n} & \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_{2\times 2} \end{bmatrix} \quad (20)$$

From (20), it can be seen that, for $i, j = 1$ to 2,

$$\boldsymbol{E}_i * \boldsymbol{D}_j = mod(\boldsymbol{E}_i * \boldsymbol{D}_j, p) = \begin{cases} \boldsymbol{I} \text{ if } j = i \\ \boldsymbol{0} \text{ if } j \neq i \end{cases} \quad (21)$$

Here, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{0}$ is the all-zero matrix of matching sizes.

Thus, a new way of generating index-wise orthogonal key matrices are derived via Householder Construction. The property represented by (21) plays an important role in the encryption and decryption process of RDSR-RJE. In (21), $\boldsymbol{E}_i$'s are the encryption keys, and $\boldsymbol{D}_j$'s are the decryption keys in Zp. All the encryption and decryption are generated by the Key Generation Center administered by the image owner. The decryption keys are sent to the respective receivers of the end users, through the secured channels.

IV. RDSR-RJE Encryption and Decryption

The architectural layout of RDSR-RJE scheme is shown in Fig. 1. The two major components are:

- RDSR-RJE Encrypter, which is also the owner of images and the corresponding RD-BM's.
- RDSR-RJE Receiver.

The RDSR-RJE Encrypter consists of the de-identification (DI) unit and the joint encryption (JEnc) unit. In RDSR-RJE, it is assumed that the ROI to be encrypted has been determined by a suitable method [9] and is made available as the Region Defining Binary Mask, ***RD-BM***, whose size is same as that of ***A***. In ***RD-BM***, the ROI pixels of ***A*** are set to ones and the non-ROI pixels to zeros.

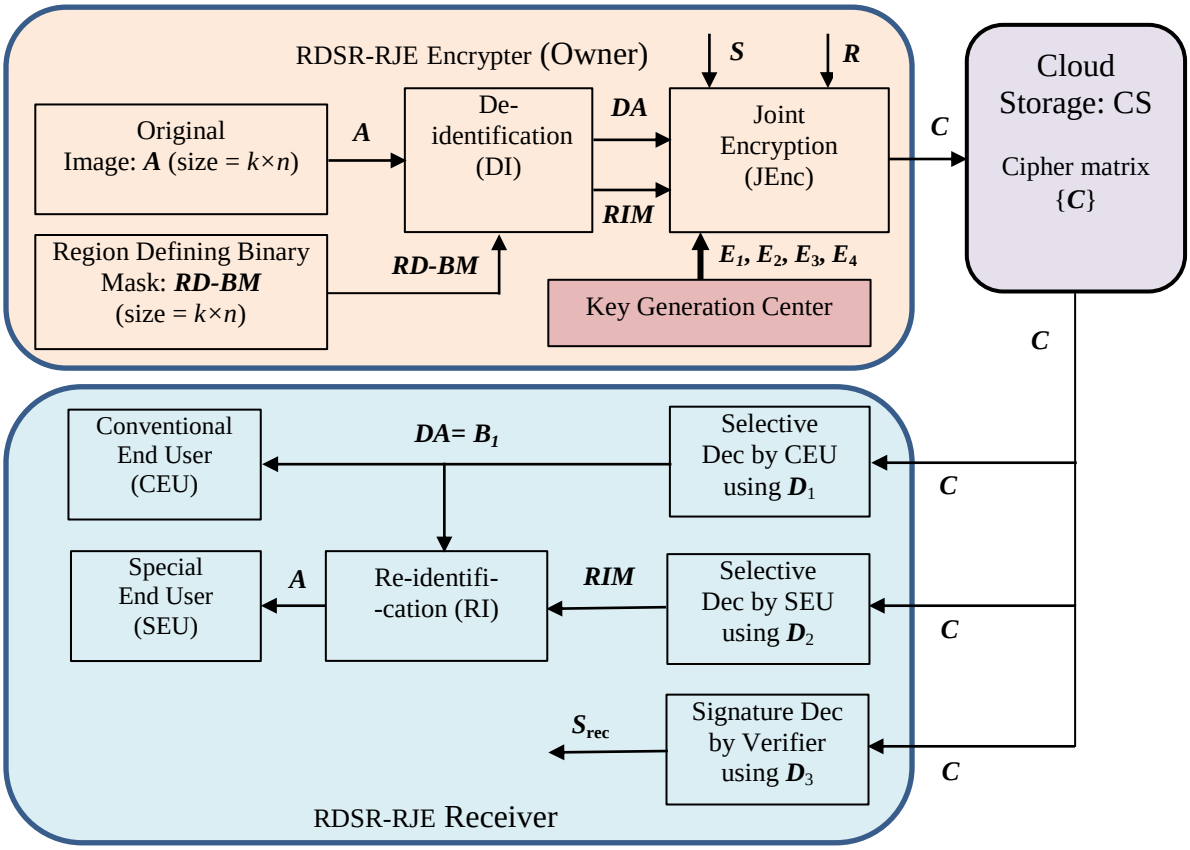


Fig. 1. The architectural layout of RDSR-RJE transmitter and receiver.

A. De-identification by Selective Encryption

The basic selective encryption process for de-identification is depicted in Fig. 2. The original input image to be de-identified is represented by matrix ***A*** of size *k*×*n* with data type uint8. (For a square image, *k* = *n*). Hereafter, when there is no ambiguity, 'image matrix ***A***' and 'image ***A***' are used synonymously.

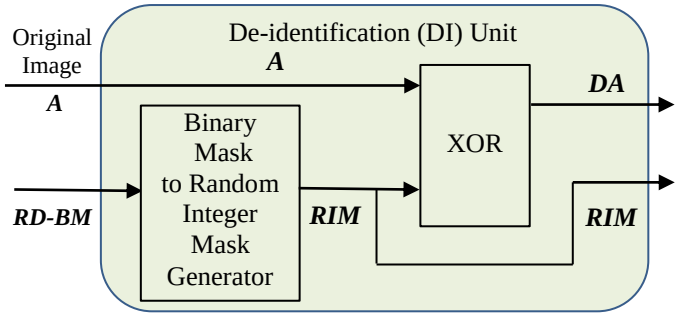


Fig. 2. The De-identification unit.

B. Binary Mask to Random Integer Mask Generator

In RDSR-RJE, the 'Binary Mask to Random Integer Mask Generator transforms the ***RD-BM*** into ***RIM***. In generating ***RIM***, the 1's of the ***RD-BM*** are replaced by random integers in the range 1-254 to get maximum diversity. Here, integer 0 is avoided as it does not make any change when XORed with any other number. Similarly, integer 255 is avoided to avoid the exact complements. The zeros of ***RD-BM*** are retained as they are in getting ***RIM***. A toy example of an 8x8 ***RD-BM*** is shown in Fig. 3(a). The corresponding ***RIM*** is shown in Fig. 3(b). The number of zeros and their locations in the ***RD-BM*** are same as in its ***RIM***. The number of integers greater than zero in ***RIM*** is equal to the number of ones of ***RD-BM***.

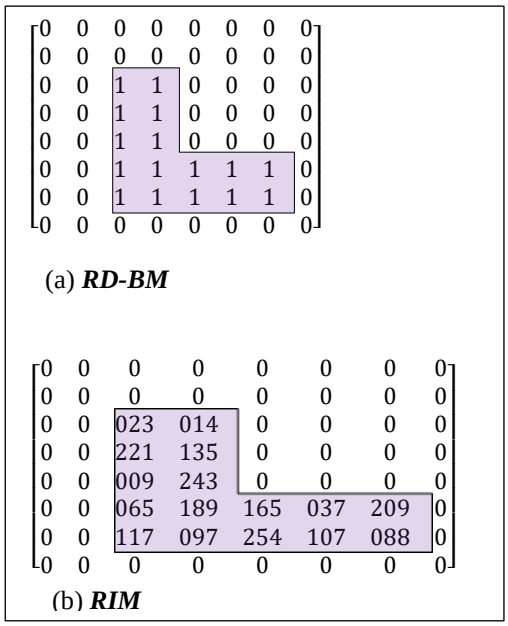


Fig. 3. RD-BM of size 8x8 and its RIM.

C. XOR De-identification

The bitwise XOR de-identification is carried out to get the de-identified out ***DA*** (De-identified ***A***) of size *k*×*n* as,

$$\boldsymbol{DA} = bitxor(\boldsymbol{A}, \boldsymbol{RIM}) \quad (22)$$

The zeros and their locations in ***RIM*** represent the non-ROI regions of ***A***. Therefore, the bitxor(…) operation does not change those pixel values. Thus the non-ROI areas of ***A*** and ***DA*** are exactly the same. On the other hand, the ROI pixel values get randomly altered at ROI locations of ***A*** due to the bitwise XOR of operations of nonzero operands. On the receiving side, the original matrix ***A*** is recovered as ***A*** = *bitxor*(***DA***, ***RIM***).

D. Security Aspects of Bitwise XOR Encryption

*1) Blind guess of **RIM**:* Let nz be the number of nonzero elements in the **RIM** used in (22). Each element can take any value between 1 and 254 (254 possibilities). Then the probability of correctly guessing a single element is (1/254), and that of correctly guessing all the elements of **RIM** is $(1/254)^{nz} = 254^{-nz}$, which is a very meager fraction, even for a moderately sized ROI.

2) Protection against chosen plain text attack (CPA): The most vulnerable attack on XOR encryption is CPA, whereby knowing **A** and **DA**, the attacker can get hold of **RIM** as,

$$\boldsymbol{RIM} = \text{bitxor}(\boldsymbol{A}, \boldsymbol{DA}) \quad (23)$$

But the ***RIM*** used in (22) is randomly varied from encryption to encryption. Here the ***RIM*** acts as an OTP. Therefore, the RIM captured by the attacker using (23) is useless for the succeeding encryptions. Thus the CPA and 'many time pad' attacks are eliminated.

*3) Accessibility and manipulation of **DA**:* In RDSR-RJE, the XOR encryption is not a standalone operation. The output matrix **DA** is internal to the sender, and it is further encrypted, as will be explained in the next section. Hence, **DA** is not directly accessible to the attacker, and any manipulation to **DA** is flagged off by the subsequent message authentication scheme.

E. Joint Encryption using Matrix Keys

Joint encryption (JEnc) is the new innovation of RDSR-JRE. Here, four data matrices are encrypted jointly with four corresponding encryption matrix keys $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ and $\boldsymbol{E}_4$ to get the cipher matrix ***C*** as,

$$\boldsymbol{C} = mod\big((\boldsymbol{DA} * \boldsymbol{E}_1 + \boldsymbol{RIM} * \boldsymbol{E}_2 + \boldsymbol{S} * \boldsymbol{E}_3 + \boldsymbol{R} * \boldsymbol{E}_4), p\big) \quad (24)$$

Here, ***S*** is the signature matrix of size *k*×*2* and ***R*** is the randomization matrix of size *k*×*2*, which is altered for successive encryptions. The sizes of ***DA*** and ***RIM*** are *k*×*n*. The sizes of encryption key matrices $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$, are *n*×*L* and the sizes of $\boldsymbol{E}_3$ and $\boldsymbol{E}_4$ are 2×*L*. The size of the cipher matrix ***C*** is *k*×*L* where *L* = 2**n* +4. Matrix ***C*** is calculated using modular algebra in Zp. Then, Eqn. (24) can be written in a simple form as,

$$\boldsymbol{C} = \boldsymbol{DA} * \boldsymbol{E}_1 + \boldsymbol{RIM} * \boldsymbol{E}_2 + \boldsymbol{S} * \boldsymbol{E}_3 + \boldsymbol{R} * \boldsymbol{E}_4 \quad (25)$$

Formation of $\boldsymbol{C}$ using (25) is called the 'Joint Encryption (JEnc)' to indicate that matrix $\boldsymbol{C}$ is obtained by encrypting four data matrices by four matrix keys to get a single weighted sum. After JEnc, the cipher matrix $\boldsymbol{C}$ is sent to the CS for secure storage and subsequent distribution. When the sizes of the matrices are large, the partial sums on the RHS may exceed the max(int) level of the computing device. In such cases, the integer overflow error can be avoided by adopting 'cumulative summation' where one term is added at a time to the partial sum followed by the mod operation, as shown below.

1) Cumulative addition: The cumulative addition is carried out as follows

C =[ ]; C=C+mod(DA*E1,p); C=mod(C,p);

C=C+mod(RIM*E2,p); C=mod(C,p);

C=C+mod(S*E3,p); C=mod(C,p);

C=C+mod(R*E4,p); C=mod(C,p);

The mod operation in each step keeps the result between 0 and (p–1). Thus the final sum $\boldsymbol{C}$ also remains within Zp.

F. Randomization Matrix $\boldsymbol{R}$

The integer matrix $\boldsymbol{R}$**,** of size $k \times n$ (same as that of $\boldsymbol{DA}$ and $\boldsymbol{RIM}$), on the RHS of (25) provides randomization to the encryption. $\boldsymbol{R}$ varies randomly from the present encryption to the next encryption. Thus, for the same inputs $\boldsymbol{DA}$, $\boldsymbol{RIM}$ and $\boldsymbol{S}$, the output $\boldsymbol{C}$ will differ in successive encryptions because of $\boldsymbol{R}$. Thus, randomized encryption is achieved, which prevents the Chosen Ciphertext Attack (CCA). The elements of R are chosen randomly from the uniform distribution in the range 0 to 255.

G. Signature Matrix $\boldsymbol{S}$

The signature matrix $\boldsymbol{S}$ of size $k \times n$ (same as that of $\boldsymbol{DA}$ and $\boldsymbol{RIM}$) provides image and source authentication. In general, Matrix $\boldsymbol{S}$ contains the ID of the source, time stamp for source authentication, and suitable hash value of the original image matrix $\boldsymbol{A}$. The source ID acts as the digital signature which is made available to the signature verifier at the receiving end.

H. Selective Decryption

The blocks used in selective decryption operation are shown in Fig. 1.

1) Decryption by a Conventional End User (CEU): The cipher matrix **C** can be decrypted by a Conventional End User (CEU) using the decryption key $\boldsymbol{D}_1$. The CEU, on decryption, recovers the de-identified image matrix DA. The decryption is carried out as,

$$\boldsymbol{B}_1 = mod(\boldsymbol{C} * \boldsymbol{D}_1, p) = \boldsymbol{C} * \boldsymbol{D}_1 \quad (26)$$

On substituting for $\boldsymbol{C}$ from (25) in (26), we get,

$$\boldsymbol{B}_1 = (\boldsymbol{DA} * \boldsymbol{E}_1 + \boldsymbol{RIM} * \boldsymbol{E}_2 + \boldsymbol{S} * \boldsymbol{E}_3 + \boldsymbol{R} * \boldsymbol{E_4}) * \boldsymbol{D}_1 = \boldsymbol{DA} * \boldsymbol{E}_1 * \boldsymbol{D}_1 + \boldsymbol{RIM} * \boldsymbol{E}_2 * \boldsymbol{D}_1 + \boldsymbol{S} * \boldsymbol{E}_3 * \boldsymbol{D}_1 + \boldsymbol{R} * \boldsymbol{E}_4 * \boldsymbol{D}_1 \quad (27)$$

From the property (21), $\boldsymbol{E}_1 * \boldsymbol{D}_1 = \boldsymbol{I}$, and the other product terms $\boldsymbol{E}_2 * \boldsymbol{D}_1, \boldsymbol{E}_3 * \boldsymbol{D}_1$ and $\boldsymbol{E}_4 * \boldsymbol{D}_1$ ara all zeros. Hence, $\boldsymbol{B}_1 = \boldsymbol{DA}$ which proves the correctness of the decryption by the CEU.

2) Selection of the modulus p: For the existence of modular inverses in Zp, the p value has to be a prime integer. In RDSR-RJE, the encryption and the decryption operations are carried out on images whose elements belong to uint8 with a maximum value = 255. Now consider the decryption, $\boldsymbol{B}_1 = mod(\boldsymbol{C} * \boldsymbol{D}_1, p)$. Here, due to the mod(…) operation, the maximum element of $\boldsymbol{B}_1$ is less than p. However, $\mathbf{B}_1$ (which is same as **DA**) represents the de-identified image where the maximum element can go up to 255. Therefore, for the correct realization of DA at the receiver, the constraint is 255 < p. That is, p should be greater than 255. The immediate higher prime number is 257, and hence **p is chosen to be 257**. Using a higher prime number for p, unnecessarily increases the cipher text size.

3) Decryption by the special end user (SEU): The special end user (SEU) has the decryption key $\boldsymbol{D}_2$ as well as $\boldsymbol{D}_1$. Using $\boldsymbol{D}_2$ the SEU decrypts **C** as,

$$\boldsymbol{B}_2 = mod(\boldsymbol{C} * \boldsymbol{D}_2, p) = \boldsymbol{C} * \boldsymbol{D}_2 \quad (28)$$

On substituting for $\boldsymbol{C}$ from (25) in (28) and again using the property (21) we get $\boldsymbol{B}_2 = \boldsymbol{RIM}$. The SEU also decrypts $\boldsymbol{C}$ using $\boldsymbol{D}_1$ to get $\boldsymbol{DA}$ as explained in section IV.*H*.1. Then the SEU can recover the original $\boldsymbol{A}$ as,

$$\boldsymbol{B} = \text{bitxor}(\boldsymbol{DA}, \boldsymbol{RIM}) \quad (29)$$

From (29) and (23), it can be seen that $\boldsymbol{B}$ is exactly equal to $\boldsymbol{A}$**,** and thus the decrypted matrix $\boldsymbol{B}$ is the exactly re-identified version of $\boldsymbol{DA}$. Therefore, RDSR-RJE achieves lossless reverse de-identification.

4) Decryption of the signature matrix $\boldsymbol{S}$*:* The signature matrix **S** is detected using the decryption key $\mathbf{D_3}$ as,

$$\boldsymbol{B}_3 = mod(\boldsymbol{C} * \boldsymbol{D}_3, p) = \boldsymbol{C} * \boldsymbol{D}_3 \quad (30)$$

On substituting for $\boldsymbol{C}$ from (25) in (30) and again using the property (21), we get $\boldsymbol{B_3} = \boldsymbol{S}$.

5) Signature verification: In RDSR-RJE, signature verification takes place before decryption. Therefore, if the verification fails, there is no need for decryption. This pre-signature verification scheme results in faster processing. The Signature Verifier (SV) at the receiving end should possess the signature decryption key $\boldsymbol{D}_3$. The SV should have already received earlier, the true signature $\mathbf{S}_{\text{true}}$ from the data owner. Now, the decrypted $\boldsymbol{B}_3$ gives the received signature $\mathbf{S}_{\text{rec}}$. Thus, on receiving **C**, the SV recovers $\mathbf{S}_{\text{rec.}}$ If $S_{rec} = S_{true}$, then the authentication is successful, and the received **C** is accepted. Otherwise, there is some error, and the present **C** is discarded, and suitable countermeasures are deployed for further investigation, or the receiver may request retransmission. Here, both CEU and SEU posses $\boldsymbol{D}_3$, and after signature verification, can proceed for further decryption. Since the signature matrix is encrypted using the encryption key $\mathbf{E}_3$, it is non-forgeable and achieves non-repudiation as the signature matrix carries the sender's ID.

I. RDSR-RJE Encryption and Decryption Algorithms

RDSR-RJE encryption and decryption algorithms are presented in this section. Modular algebra in Zp is used for all calculations.

Algorithm **RDSR-RJE encrypt**

Inputs: Original image ***A*.** Mask matrix ***RD-BM*.** Signature matrix ***S***. Encryption keys $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ and $\boldsymbol{E}_4$.
Output: Cipher matrix ***C***.

1. Get the Random Integer Matrix, ***RIM*** from **RD-BM** as explained in section IV.*A*.1.
2. Get the de-identified image matrix ***DA*** by the XOR encryption, as given by (22).
3. Generate the randomization integer matrix ***R*** using any standard library function from python, C++, Java, etc.
4. Jointly encrypt ***DA***, ***RIM***, ***S,*** and ***R*** to get the cipher matrix ***C***, using the Encryption keys $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$, and $\boldsymbol{E}_4$ as given by (24).
5. Over.

Decryption Algorithm used by the conventional end user (CEU) is summarized as follows.

Algorithm **RDSR-RJE-CEU decrypt**

Inputs: Cipher matrix ***C***. Decryption keys $\boldsymbol{D}_1$ and $\boldsymbol{D}3$. True signature matrix $\boldsymbol{S}_{true}$.
Output: De-identified image matrix ***DA*.**

1. Get $\boldsymbol{S}_{rec}$ using (30) as, $\boldsymbol{S}_{rec} = \boldsymbol{C} * \boldsymbol{D}_3$.
2. Verify the signature as:
   If $\boldsymbol{S}_{rec} \neq \boldsymbol{S}_{true}$ //Signature failure
   Discard the present ***C***, and request for Retransmission.
   Go to step 4.
   Else //Signature OK
   Go to step 3.
   Endif
3. Get ***DA*** using (26), as $\boldsymbol{DA} = \boldsymbol{C} * \boldsymbol{D}_1$.
4. Over.

RDSR-RJE decryption by the Special End User is an extension of the algorithm **RDSR-RJE-CEU decrypt.**

Algorithm **RDSR-RJE-SEU decrypt** by the Special End User

Inputs: Cipher matrix ***C***. Decryption keys $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}3$. **Output:** Original image matrix ***A*.**

1. Get ***DA*** according to the algorithm **RDSR-RJE-CEU**
2. Get ***RIM*** using (28), as $\boldsymbol{RIM} = \boldsymbol{C} * \boldsymbol{D}_2$.
3. Get the original re-identified matrix ***A***, using (29) as, ***A*** = bitxor(***DA***, ***RIM***).
4. Over.

J. Characteristics of RDSR-RJE Encryption/Decryption

1) Ciphertext expansion ratio: The Ciphertext Expansion Ratio (CER) is the ratio of the size of the cipher matrix to that of its plain matrix. For a given plain matrix, higher the value of CER, higher is the size of its cipher matrix, and consequently, the computational and communication cost becomes relatively higher. A lower CER value contributes to a higher degree of encryption efficiency. CER is defined as,

$$\text{CER} = \frac{\text{Size of Ciphermatrix inbits}}{\text{Size of Plain matrix in bits}} = \frac{\text{Size of } \boldsymbol{C} \text{ in bits}}{\text{Size of } \boldsymbol{A} \text{ in bits}}$$

In RDSR-RJE, the uppermost value of an element in the cipher matrix is ***C*** is (p–1). Hence, the number of bits needed to represent an element of ***C*** is ceil($\log_2(p–1)$) bits. With p = 257, ceil($\log_2(p–1)$) = 8. Thus, 8 bits are required to represent each element of ***C***. Now, the size of ***C*** (No. of elements in ***C***) is $k \times L$. Hence the total size of ***C*** in bits is $k*L*8$. The plain matrix has a bit depth of 8, and its size is $k \times L$. Therefore, the total size of ***A*** (or ***DA***) in bits is k*n*8. Hence,

$$\text{CER} = \frac{k*L*8}{k*n*8} = \frac{L}{n} \tag{31}$$

On substituting for L from (17), and when n is large compared to 2,

$$CER = \frac{L}{n} = \frac{2*n+4}{n} \cong 2 \tag{32}$$

An important characteristic of RDSR-RJE is that the *CER* value is constant and does not increase with *n*.

2) Lossless reversible de-identification: In RDSR-RJE, the decrypted image is the exact replica of the original image for the conventional end user or the special end user. Thus, it is a zero-loss scheme.

Additionally, RDSR-RJE does not use block-wise operations. It avoids floating point operations and iterative procedures. Therefore, RDSR-RJE is efficient and achieves higher execution speed.

K. Security Aspects of RDSR-RJE

1) Exhaustive search for keys: Each element of an encryption or decryption key belongs to Zp whose range is 0 to (p–1). Therefore an element of a key can take any one value out of p possibilities. Hence the probability of correctly guessing a single element is (1/p). Each key has n*L elements. Therefore, the probability of guessing all the elements correctly is $(1/p)^{n*L} = p^{-n*L}$ which is extremely a very low value for p = 257. Thus, the success of an exhaustive search is negligibly small.

2) Protection against chosen plain text attack (CPA): In RDSR-RJE, the encryption process is randomized using the random matrix **R** as in (25) where **R** is varied from encryption to encryption. Thus, the cipher matrix would be different even if the input plain matrix is same for consecutive encryptions. Hence, the randomized encryption prevents CPA.

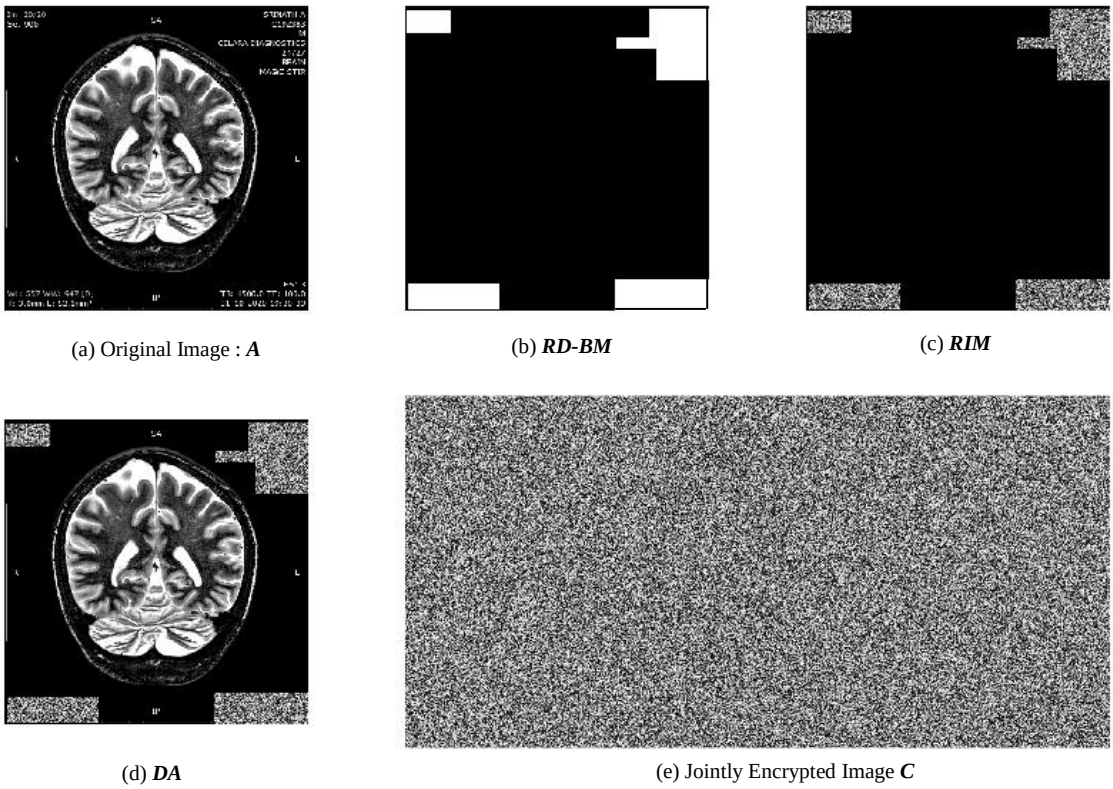
V. Experimental Results and Discussion

A. Experiment 1

An axial MRI view is taken as the original grays scale plain image as shown in Fig. 4(a). The image matrix ***A*** is of size 512×512. The regions selected for de-identification are the textual details that include the patient's name, date of image production *etc*. The selected regions are shown in the ***RD-BM*** of Fig. 4(b). The corresponding Random Image Mask (***RIM***) is shown in Fig. 4(c). The de-identified (selectively encrypted) image ***DA***, as given by (22), is shown in Fig. 4(d). The result of JEnc, matrix ***C*** obtained using (25) is shown in Fig. 4(e) which shows a very high degree of randomness.

*1) Histograms of **DA** and **C**:* Histograms of the de-identified image **DA** and the jointly encrypted cipher image **C** are shown in Fig. 5. From Fig. 5, it can be seen that the histogram of **C** is uniformly distributed compared to that of **DA**. This shows the comprehensive randomness of **C**. A malicious attacker cannot deduce any information from the histogram of **C**.

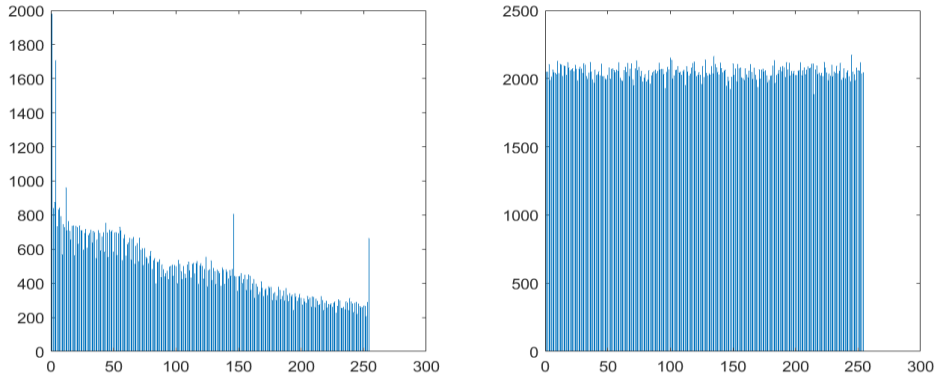
2) Comparison of visual correlation coefficients: Adjacent pixel values of a normal image, have a higher correlation between them where as in a good cipher image, the corresponding correlation should be very low. That means, in a cipher image, the adjacent pixel values are highly dispersed. Hence the correlation coefficient [29] will below.



(a) Original Image : ***A*** (b) ***RD-BM*** (c) ***RIM***

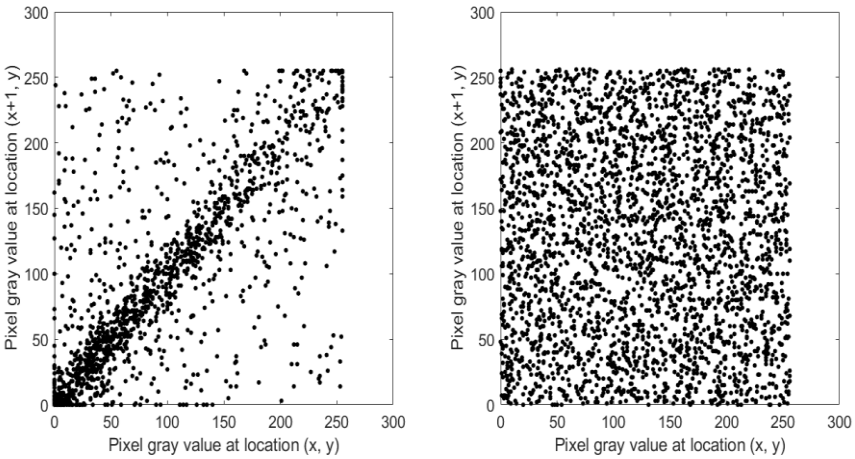
(d) ***DA*** (e) Jointly Encrypted Image ***C***

Fig. 4. Original Image A, RD-BM, RIM and the encrypted image C (Experiment 2).



(a) Histogram of ***DA*** (b) Histogram of ***C***

Fig. 5. Histograms of DA and C.



(a) De-identified image (b) Jointly encrypted cipher

Fig. 6. Comparison of correlation coefficients between DA and C.

Comparison of pixel value dispersions along horizontal direction, between ***DA*** and the cipher image ***C*** of Experiment 1, is shown in Fig. 6, from which it can be seen that the cipher image ***C*** has a high degree of pixel value dispersion compared to ***DA***.

B. Experiment 2

In this experiment, an ultrasound scan of pregnancy is the original image ***A*** with size 187×269, as shown in Fig. 7(a). The region selected for de-identification is the central part of the ***A*** as shown by the binary mask, ***RD-BM*** shown in Fig. 7(b), and the corresponding ***RIM*** is shown in 7(c). The de-identified (selectively encrypted) image ***DA***, as given by (22), is shown in Fig. 7(d). The cipher matrix ***C*** obtained using (25) is shown in Fig. 7(e), which shows a very high degree of randomness.

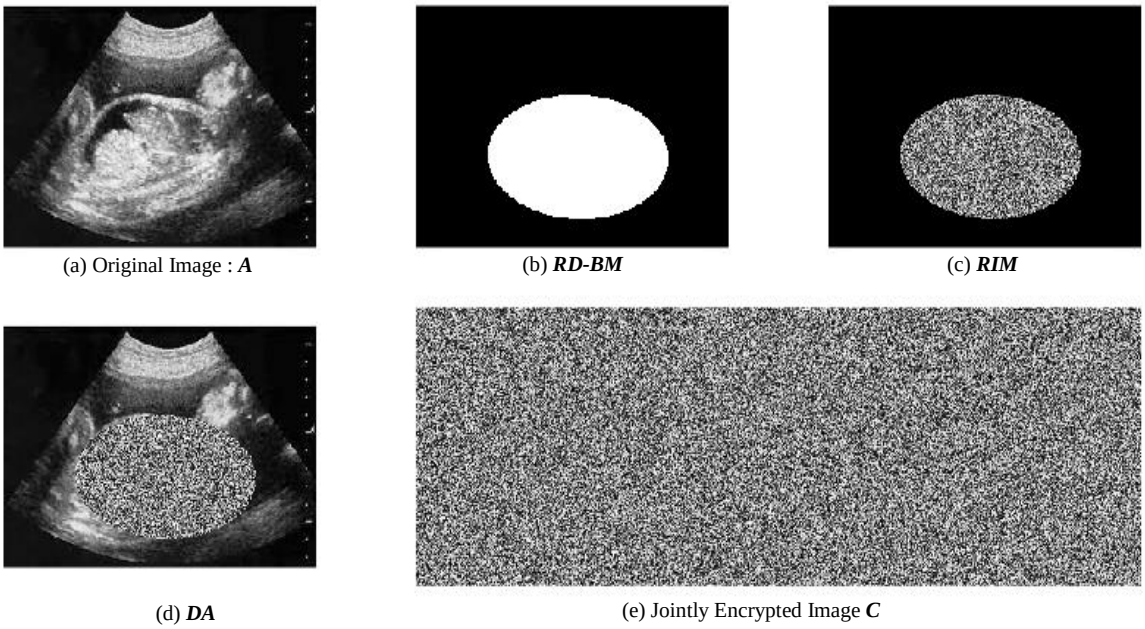
C. Metrics for Comparison

A few metrics for comparing the image encryption schemes are discussed in this section.

1) Differential analysis: Differential analysis is the study of the variations in the cipher matrix when the plain matrix changes by a small value. Thus, it is basically a sensitivity analysis. A quantitative measure of this behavior is NPCR which stands for the Number of Pixels Change Rate.

Let $\boldsymbol{C}_1$ be the cipher image of a given plain image. Let $\boldsymbol{C}_2$ be the resulting cipher image after a one-bit change in the plain image. The differential change per pixel is defined as,

$$d(i,j) = \begin{cases} 1 \text{ if } C_1(i,j) \neq C_2(i,j) \\ 0 \text{ otherwise} \end{cases} \quad (33)$$



(a) Original Image : ***A*** (b) ***RD-BM*** (c) ***RIM***

(d) ***DA*** (e) Jointly Encrypted Image ***C***

Fig. 7. Original Image A, RD-BM, RIM, DA and the encrypted image C (Experiment 2).

for i = 1 to h and j = 1 to w where h and w are the height and width of the cipher image. Then, the percentage NPCR is defined [41] as,

$$NPCR = \frac{\sum_{i=1}^{h}\sum_{j=1}^{w} d(i,j)}{h*w} * 100 \quad (34)$$

A higher *NPCR* means the attacker cannot capture the encryption due to the large number of indeterminates. Thus, higher the NPCR, higher is the security of the encryption process. The ideal value of the NPCR is 100%. Another metric that measures the differential score is *UACI* (unified averaged changed intensity) which is defined [41] as,

$$UACI = \frac{\sum_{i=1}^{h}\sum_{j=1}^{w} | C_1(i,j) - C_2(i,j)|}{h * w * 256} * 100 \quad (35)$$

Higher the value of UACI, better is the encryption performance.

2) Image entropy: The entropy of an image in bits/pixel, is defined as,

$$H = -\sum_{i=0}^{i=imax} p_i * log_2(p_i) \quad (36)$$

where p_i is the probability of a pixel having the gray level *i*, and *imax* is the maximum gray level (255 in a normal image). For a fully random image, H = 8 bits/pixel.

3) Structural similarity index: The structural similarity index (SSIM) [30] measures the closeness between two images. In RDSR-RJE the de-identified image **DA** and its decrypted version $\mathbf{B}_1$ are exactly same. Hence SSIM value in RDSR-RJE is 1. SSIM values less than 1 imply recovery of the plain image with error.

D. Comparison of the Performance of RDSR-RJE

The encryption efficiency parameters of RDSR-RJE are compared with those of HOSSAIN [28], QIN [29], and JITHIN [30]. The numerical results are shown in Table I, for images 'Img 1' and 'Img 2' which are from Fig. 4(a) and 7(b), respectively.

1) Execution time: The theoretical time complexity calculations of the different methods [28-30] are extensive and depend on the respective contexts. Therefore, the execution times of the encryption algorithms are obtained experimentally and shown in the plots of Fig. 8. Here, the image used in Experiment 1 is resized starting from 64x64 and progressively increased upto 512x512 as marked in Fig. 7. Then the corresponding execution times are calculated using the appropriate Matlab code. In Fig. 8, the execution time of joint encryption by RDSR-RJE, is shown in black.

The values obtained in Fig. 8 are machine-dependent, and thus the execution times are relative only. From Fig. 8, it can be seen that RDSR-RJE has a significant lower execution time compared to the other three methods. For example, when the image size is 256x256, the percentage improvement in the execution time of RDSR-JE compared to that of HOSS [28] is, (30.38–21.35)*100/30.38, which is approximately equal to 30%.

TABLE I. COMPARISON OF THE QUANTITATIVE VALUES OF THE METRICS

	Plain Image	**Encrypted Image**			
Horizontal Correlation Coefficient					
		RDSR-RJE	HOSS [28]	QIN [29]	JITHIN [30]
Img 1	0.8772	0.0025	0.0027	0.0029	0.0037
Img 2	0.7421	0.0227	0.0312	0.0412	0.0467
Image Entropy					
Img 1	4.8876	7.9974	7.9948	7.9953	7.9916
Img 2	7.2674	7.9961	7.9943	7.9951	7.9919
Number of Pixels Change Rate (NPCR) in percentage					
Img 1	----	99.6037	99.5100	99.4991	99.4211
Img 2	----	99.5965	99.4235	99.4173	99.4053
Unified Averaged Changed Intensity (UACI) in percentage					
Img 1		33.5239	32.9932	32.4327	32.2327
Img 2		33.2405	32.0159	32.0079	31.8953
Structural Similarity Index (SSIM)					
Img 1		1.00	0.9879	0.9752	0.9623
Img 2		1.00	0.9693	0.9533	0.9457

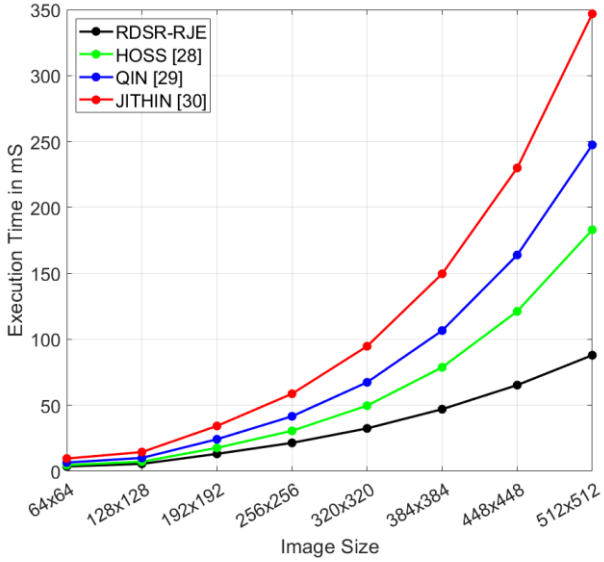


Fig. 8. Comparison of execution times for image encryption.

VI. CONCLUSION

A new method of reversible image de-identification by encryption has been presented. It uses matrix keys for asymmetric encryption and decryption of image matrices. Content and source authentication via a digital signature scheme is integrated using joint encryption. All the cryptographic operations are carried out in the finite field Zp and thus avoid the floating point operations that lead to higher computational speed. Moreover, the algorithm is non-iterative and does not use block-wise operations to achieve faster results. Here, the decrypted image is the exact replica of the original image, and thus, it is a zero-loss scheme. Additionally, the encryption/decryption security-related performance parameters, namely, entropy, correlation coefficients, NPCR, and UACI, are very near to their ideal values. The proposed method, on average, reduces the execution time of homomorphic encryption by 30 to 40 percent.

REFERENCES

[1] Raj, B.S.S., Venugopalachar, S. Multi-data Multi-user End to End Encryption for Electronic Health Records Data Security in Cloud. Wireless Pers Commun 125, 2413–2441 (2022). https://doi.org/10.1007/s11277-022-09666-2.

[2] Hossein Ghanbari-Ghalehjoughi, Mansour Eslami, Sohrab Ahmadi-Kandjani, Mohsen Ghanbari-Ghalehjoughi, Zeyun Yu, Multiple layer encryption and steganography via multi-channel ghost imaging, Optics and Lasers in Engineering, Volume 134, 2020, 106227, ISSN 0143-8166, pp. 1-12.

[3] Kaur, M., Kumar, V. A Comprehensive Review on Image Encryption Techniques. Arch Computat Methods Eng **27**, 15–43 (2020). https://doi.org/10.1007/s11831-018-9298-8.

[4] A.S. Sajitha, A. Shobha Rekh, "Review on various image encryption schemes, Materials Today: Proceedings, Volume 58, Part 1," 2022, Pages 529-534, doi: 10.1016/j.matpr.2022.03.058.

[5] K. Suneja, S. Dua and M. Dua, "A Review of Chaos-based Image Encryption," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, pp. 693-698, doi: 10.1109/ICCMC.2019.8819860.

[6] Jameel, E. A., & Fadhel, S. A. (2022). Digital Image Encryption Techniques: Article Review. Technium: Romanian Journal of Applied Sciences and Technology, *4*(2), pp. 24–35. Doi: 10.47577/technium.v4i2.6026.

[7] C. Tiken and R. Samlı , "A Comprehensive Review About Image Encryption Methods", Harran Üniversitesi Mühendislik Dergisi, vol. 7, no. 1, pp. 27-49, Apr. 2022, doi:10.46578/humder.1066545.

[8] G.S. Nelson, Practical Implications of Sharing Data: A Primer on Data Privacy, Anonymization, and De-Identification (2015), pp. 1–23 ⟨http://thotwave.com/wpcontent/uploads/2015/09/ data_sharing privacy anonymization_ and_deidentification_rev_13. pdf⟩ (accessed 25.06. 15).

[9] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing Using MATLAB", (Chapters 11 and 12), Third Edition, Gatesmark Publishing®. A Division of Gatesmark, ® LLC, [Knoxville], Tennessee, USA, 2020.

[10] Jawad, Lahieb & Sulong, Ghazali. (2015). A Survey on Emerging Challenges in Selective Color Image Encryption Techniques. Indian Journal of Science and Technology. 8. pp. 1-13. doi:10.17485/ijst/2015/v8i27/71241.

[11] Slobodan Ribaric, Aladdin Ariyaeeinia, Nikola Pavesic, "De-identification for privacy protection in multimedia content: A survey," Signal Processing: Image Communication, Volume 47, 2016, Pages 131-151, https://doi.org/10.1016/j.image.2016.05.020.

[12] A. J. Paul, "Recent Advances in Selective Image Encryption and its Indispensability due to COVID-19," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2020, pp. 201-206, doi: 10.1109/RAICS51191.2020.9332513.

[13] Khan, Jan Sher & Ahmad, Jawad. (2019). Chaos based efficient selective image encryption. Multidimensional Systems and Signal Processing. 30. doi: 10.1007/s11045-018-0589-x.

[14] Cun, Q., Tong, X., Wang, Z., & Zhang, M. (2021). Selective image encryption method based on dynamic DNA coding and new chaotic map. Optik, *243*, *167286*. pp. 1-29. doi:10.1016/j.ijleo.2021.167286.

[15] Kiran and Parameshachari B. D. "Selective Image Encryption of Medical Images Based on Threshold Entropy and Arnold Cat Map, " Biosc.Biotech.Res.Comm. Special Issue Vol 13. No 13. 2020, pp. 194-202. doi: 10.21786/bbrc/13.13/27.

[16] Mehmet Yamac, et al. "Reversible Privacy Preservation using Multi-level Encryption and Compressive Sensing," arXiv:1906.08713v1 [cs.CR], EUSIPCO 2019, pp. 1-5. doi: 10.48550/ arXiv. 1906.08713.

[17] Carrillo, Paula & Kalva, Hari & Magliveras, Spyros. (2009). Compression Independent Reversible Encryption for Privacy in Video Surveillance. EURASIP Journal on Information Security. 2009. pp. 1-13. doi: 10.1155/2009/429581.

[18] Li, J., Zhang, Z., Li, S. et al. A partial encryption algorithm for medical images based on quick response code and reversible data hiding technology. BMC Med Inform Decis Mak 20 (Suppl 14), 297 (2020). pp. 1-16. doi: 10.1186/s12911-020-01328-2.

[19] Chuan Qin , Zhihong He , Xiangyang Luo , Jing Dong , Reversible Data Hiding in Encrypted Image with Separable Capability and High Embedding Capacity, Information Sciences (2018), doi: 10.1016/j.ins.2018.07.021.

[20] A. Hafsa, J. Malek and M. Machhout, "An Improved Security Approach for Medical Images and Patients' Information Transmission," 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2022, pp. 19-24, doi: 10.1109/SETIT54465.2022.9875481.

[21] P. Parida, C. Pradhan, X. -Z. Gao, D. S. Roy and R. K. Barik, "Image Encryption and Authentication With Elliptic Curve Cryptography and Multidimensional Chaotic Maps," in IEEE Access, vol. 9, pp. 76191-76204, 2021, doi: 10.1109/ACCESS.2021.3072075.

[22] X. Li, D. Xiao, H. Mou, D. Lu and M. Peng, "A Compressive Sensing Based Image Encryption and Compression Algorithm With Identity Authentication and Blind Signcryption," in IEEE Access, vol. 8, pp. 211676-211690, 2020, doi: 10.1109/ACCESS.2020.3039643.

[23] T. S. Ali and R. Ali, "A Novel Medical Image Signcryption Scheme Using TLTS and Henon Chaotic Map," in IEEE Access, vol. 8, pp. 71974-71992, 2020, doi: 10.1109/ACCESS.2020.2987615.

[24] R. I. Abdelfatah, "Secure Image Transmission Using Chaotic-Enhanced Elliptic Curve Cryptography," in IEEE Access, vol. 8, pp. 3875-3890, 2020, doi: 10.1109/ACCESS.2019.2958336.

[25] G. Luan, A. Li, D. Zhang and D. Wang, "Asymmetric Image Encryption and Authentication Based on Equal Modulus Decomposition in the Fresnel Transform Domain," in IEEE Photonics Journal, vol. 11, no. 1, pp. 1-7, Feb. 2019, Art no. 6900207, doi: 10.1109/JPHOT.2018.2886295.

[26] D. Reilly and L. Fan, "A Comparative Evaluation of Differentially Private Image Obfuscation," 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2021, pp. 80-89, doi: 10.1109/TPSISA52974.2021.00009.

[27] N. Tsafack, J. Kengne, B. Abd-El-Atty, A. M. Iliyasu, K. Hirota, and A. A. A. EL-Latif, "Design and implementation of a simple dynamical 4-d chaotic circuit with applications in image encryption," Information Sciences, vol. 515, pp. 191–217, apr 2020.

[28] M. B. Hossain, M. T. Rahman, A. B. M. S. Rahman and S. Islam, "A new approach of image encryption using 3D chaotic map to enhance security of multimedia component," 2014 International Conference on Informatics, Electronics & Vision (ICIEV), 2014, pp. 1-6, doi: 10.1109/ICIEV.2014.6850856.

[29] Q. Qin, Z. Liang, S. Liu, X. Wang and C. Zhou, "A Dual-domain Image Encryption Algorithm Based on Hyperchaos and Dynamic Wavelet Decomposition," in IEEE Access, 2022, doi: 10.1109/ACCESS.2022.3212145.

[30] Jithin, K. C., & Sankar, S. (2020). Colour image encryption algorithm combining, Arnold map, DNA sequence operation, and a Mandelbrot set. Journal of Information Security and Applications, 50, 102428. doi:10.1016/j.jisa.2019.102428.

[31] Z. Hua, Y. Zhou, and H. Huang, "Cosine-transform-based chaotic system for image encryption," Information Sciences, vol. 480, 2019. pp. 403–419. doi: org/10.1016/ j.ins.2018.12.048.

[32] Xingyuan Wang, Suo Gao, "Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network," Information Sciences, Volume 539, 2020, pp. 195-214, doi: 10.1016/j.ins.2020.06.030.

[33] Xiong, Fu & Xiao, Yang & Cao, Zhi-Guo & Gong, Kaicheng & Zhiwen, Fang & Zhou, Joey. (2019). Good practices on building effective CNN baseline model for person re-identification. 145. 10.1117/12.2524386.

[34] Jeong Y, Yoo S, Kim Y, Shim W, "De-Identification of Facial Features in Magnetic Resonance Images: Software Development Using Deep Learning Technology," J Med Internet Res 2020;22(12):e22739. doi: 10.2196/22739.

[35] Monteiro, Eriksson & Costa, Carlos & Oliveira, José. (2017). A De-Identification Pipeline for Ultrasound Medical Images in DICOM Format. Journal of Medical Systems. 41. 89. 10.1007/s10916-017-0736-1.

[36] Tekli, Jimmy & AL Bouna, Bechara & Couturier, Raphaël & Tekli, Gilbert & Zein, Zeinab & Kamradt, Marc. (2019). A Framework for

Evaluating Image Obfuscation under Deep Learning-Assisted Privacy Attacks. 1-10. 10.1109/PST47121.2019.8949040.

[37] Jingjing Yang, Weijia Zhang, Jiaxing Liu, Jinzhao Wu, Jie Yang, "Generating De-identification facial images based on the attention models and adversarial examples," Alexandria Engineering Journal, Volume 61, Issue 11, 2022, pp. 8417-8429, https://doi.org/10.1016/j.aej.2022.02.007.

[38] Stewart, G. W. "The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators." SIAM Journal on Numerical Analysis, vol. 17, no. 3, 1980, pp. 403–409. *JSTOR*, www.jstor.org/stable/2156882.Accessed 22 Mar. 2021.

[39] Jacques-García, Fausto & Uribe-Mejía, Daniel & Macías-Bobadilla, Gonzalo & Chaparro-Sánchez, Ricardo. (2019). On modular inverse matrices A computational approach. South Florida Journal of Development, Miami, vol.3, no.3. pp.3100-3111, 2019.

[40] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "Handbook of Applied Cryptography (last updated July 8, 2011)," CRC Press, Inc. Boca Raton, FL, USA ©1996.

[41] Wu, Yue. (2011). "NPCR and UACI Randomness Tests for Image Encryption," Cyber Journals: Journal of Selected Areas in Telecommunications. April, 2011. pp.31-38.