Securing Privacy with Matrices: A New Approach to Image Data Protection

Mohammed Abbas Fadhil Al-Husainy[1]

[1] Department of Cyber Security, Faculty of Science and Information Technology, Irbid National University, Irbid, Jordan.

***Abstract*— As the digital landscape evolves, protecting sensitive image data becomes paramount in preserving user privacy. This paper introduces a novel approach for safeguarding image data using matrix-based encryption techniques. Traditional encryption methods may not be suited for images due to their unique characteristics, such as large data sizes and complex visual patterns. In response to this challenge, our proposed method exploits the power of matrices to effectively transform and secure image data. In this study, we introduce an encryption method for an image matrix by performing several matrix manipulations that include permuting rows with columns, shifting elements in rows and columns, and XORing elements in the main and secondary diagonals with a randomly generated key. Extensive experiments were conducted on various images using different security and performance metrics to assess the effectiveness of the suggested method. The results showed that the presented matrix-based encryption method competes with known methods regarding encryption strength and attack resistance. This research provides a promising encryption method that uses the matrix model to achieve security for image data.**

***Keywords*— Privacy Protection, Image Data Security, Matrix Approach, Encryption**

I. INTRODUCTION

In the era of continuous progress and controlled technology, the proliferation of image data has become an essential part of our daily lives. Images, whether they are personal photos, medical scans,

surveillance footage, or artistic creations, constitute a rich, confidential, and complex form of data [1]. The development of several encryption techniques has been driven by the need to protect the privacy and integrity of these images from unauthorized access or manipulation. [2]. Matrices are a type of data structures and mathematical constructs that provide a versatile framework that can be used to transform and manipulate data. Most encryption techniques use matrices to represent the data to be encrypted, S-boxes, and to represent keys [3].

Permutation and transposition operations have long been considered essential parts of cryptographic processes because they can effectively break data patterns and render them incomprehensible to unauthorized parties. It can take advantage of the mathematical properties of matrices to seamlessly apply permutation and transposition operations to image data. These operations form the basis of the encryption system and provide robust protection against different threats [1], [4].

The purpose of this research article is to study the principles, techniques, and applications of using matrices in image encryption. We will explore the theoretical foundations of matrix-based image encryption, examine its implementation, and evaluate and discuss the results to be obtained.

The rest of this article is structured as follows: Section 2 provides a detailed overview of existing approaches related to matrix-based image encryption. In Section 3, the proposed methodology for using matrices to encrypt images is introduced. Section 4 describes the implementation of the suggested method and discusses the evaluation of its performance compared with the known encryption. Finally, Section 5 concludes this article with a summary of findings and recommendations regarding future directions in this field.

II. RELATED WORKS

Xian, Y., and Wang, X. [5] proposed a type of sorting matrix with fractal features called the fractal sorting matrix (FSM) and developed an iterative calculation method for this matrix to make it self-similar, irregular, and infinitely recursive. Moreover, to achieve high encryption efficiency and better security, the authors presented a new technique for global pixel diffusion with two chaotic sequences. Comparative experiments showed that the suggested encryption method is faster, resistant to a variety of attacks, and provides better security.

DOI:http://doi.org/10.24086/cocos2024/paper.1509

A Boolean network coding algorithm for synchronous updating processes is proposed by Wang, X., and Gao, S. In addition to studying chaotic image encryption techniques, we are also exploring the application of matrix semi-tensor product theory. A random key stream is created using a 2D-LASM chaotic model. A Boolean matrix is first created after coding a Boolean network. If necessary, the logical network matrix is diffused in a single round and can then be saved as an image. The plaintext image is then encoded using three random positions scrambling. Finally, a new Boolean network can be created by encrypting the encoded image using the matrix semi-tensor product approach and a second round of diffusion. The algorithm showed a strong security advantage when compared to other algorithms [6].

Because both bit-level alteration and pixel-level alteration used in image encryption methods have actual drawbacks. To overcome these drawbacks, Zhang, W., et al. suggested a new cryptosystem. First, a comprehensive examination and comparison of several permutation algorithms are performed. The Chen model is used to generate a bit-level random visit mechanism of a plain image. From a bit-level point of view, an image can be represented as a 3D array containing width, height, and bit depth. A new mapping rule has been created to randomly map one position to another within a 3D matrix. This method involves a double random position permutation, instead of the traditional sequential visit of the plain image. It combines elements of the Chen system and the 3D Cat map in the permutation stage [7].

Hu, X. et al. introduced a novel complex chaotic system that generates dynamic random variations of chaotic sequences through the integration of the cloud model and the generalized Fibonacci model. The pixel coordinates of the mosaic images of the R, G, and B components of the color image are randomly arranged in a chaotic order. Then, apply the chaotic sequence value as a matrix convolutional cloud method that alternately updates the input value of the matrix convolution operation and the pixel value to obtain the alteration transformation of the pixel value origin. Finally, by applying the chaotic sequence value as the matrix convolution cloud method, the input value of the matrix convolution operation and the pixel value are alternately updated to obtain the permutation transformation of the original pixel value. Evaluation tests showed that the encoded image's histogram is smoother and its nearby pixels exhibit low correlation. This proposed system provides great robustness, a high level of encryption security, and anti-interference. It is also immune against select plaintext attacks, differential attacks, and noise attacks [8].

Quantum matrix-based image encryption has gained popularity as a result of the development of quantum computing. Quantum matrix operations were studied by Zhang et al. for image encryption, providing improved security through quantum key distribution [9].

III. Proposed Matrix-based Image Encryption

To ensure that attackers cannot extract useful information from the encrypted image and protect it against attacks, a high distortion rate must appear in the encoded image. The desired distortion in the encoded image can be achieved by selecting a combination of efficient permutation and transposition operations to ensure the necessary confusion and diffusion effects occur in the encrypted image.

A. Preparation Stage

The goal of the suggested image encoding method is to use the 2D matrix structure to represent image data and perform the necessary matrix operations to create a highly secure encrypted image.

Initially, the original image *S* is represented as a square matrix *M* by reading the image data byte by byte. Processing image data byte by byte rather than pixel by pixel deepens the effect of applying encryption operations to the source image. The dimension M_{Dim} of the square matrix is calculated using (1) and (2).

$$S_{Size} = \text{Width} \times \text{Height} \times \text{Palette} \quad (1)$$

$$M_{Dim} = \sqrt[2]{S_{Size}} \quad (2)$$

A Random Number Generation Algorithm (RNGA) is adopted and kept secret for use in the suggested image encoding method. When implementing the suggested image encryption method, multiple seeds are passed to RNGA to increase the randomness of the number sequence generated by RNGA.

B. Functions Implemented on the Matrix M

A set of functions has been written to be performed on the matrix *M* during the encryption stage. These functions are classified into two groups:

Group 1 - (Permutation Functions): The logical XOR (exclusive OR) operation is chosen in this work to implement the permutation operation on the bytes in *M*. Choosing of logical XOR operation instead of mathematical operations aims to reduce encryption time. The XOR operation involves taking two binary inputs and producing a binary output.

In each of the following functions, a vector *V* of length M_{Dim} contains random bytes generated by the RNGA. All these functions use different seed values for the RNGA. A list of necessary random seed values is generated at the beginning of the encryption stage.

1) XORRows: the function performs an XOR logic operation between each row in *M* and a different, randomly generated vector *V*.

2) XORColumns: the function performs an XOR logic operation between each column in *M* and a different, randomly generated vector *V*.

3) XORMainDiagonal: the function performs an XOR logic operation between the main diagonal of *M* and a randomly generated vector *V*.

4) XORSecondaryDiagonal: the function performs an XOR logic operation between the secondary diagonal of *M* and a randomly generated vector *V*.

Group 2 – (Transposition Functions): These functions include swap operations between bytes at different indices in *M*, rotate operations for bytes in rows (left/right), and columns

DOI:http://doi.org/10.24086/cocos2024/paper.1509

(up/down). Most of these functions use different seed values for the RGA. A list of necessary random seed values is generated at the beginning of the encryption stage.

SwapAboveUnderMainDiagonal: the function swaps the bytes above the main diagonal with the bytes under the main diagonal in *M*.

SwapAboveUnderSecondaryDiagonal: the function swaps the bytes above the secondary diagonal with the bytes under the secondary diagonal in *M*.

SwapTwoRows: the function swaps the bytes in each row in *M* with the bytes in another row in *M*. The number of the second row is randomly determined.

SwapTwoColumns: the function swaps the bytes in each column in *M* with the bytes in another column in *M*. The number of the second column is randomly determined.

RotateRowRight: the function rotates the bytes in each row in *M* to the right at a different, randomly generated offset.

RotateRowLeft: the function rotates the bytes in each row in *M* to the left at a different, randomly generated offset.

RotateColumnUp: the function rotates the bytes in each column in *M* up at a different, randomly generated offset.

RotateColumnDown: the function rotates the bytes in each column in *M* down at a different, randomly generated offset.

C. Encryption Stage

After providing a demonstration of the functions that will be used to implement matrix operations on the image data matrix *M*, the encryption stage of the suggested image encryption method involves using the RNGA to obtain a random sequence of calls to these functions.

A list of all random seed values needed to implement all functions is also generated here.

The user must choose a digital file, regardless of its type and size, to use as a key *K*. The key *K* is treated as a sequence of bytes and is used to calculate the seed value of the RNGA in the encryption stage. The seed value is calculated using (3), where K_{Size} is the size of the key file in byte.

$$Seed = XOR_{i=1}^{K_{Size}} K[i] \quad (3)$$

IV. Experiments and Evaluation Metrics

Several metrics were used to assess the effectiveness of the proposed image encryption method and to ensure that utilizing the matrix and its operations for encrypting images results in a high level of security for the encrypted images. Different colored images of varying sizes were utilized in the experiments and some of these images are listed in Fig. 1.

The following subsections describe some of the performance evaluation metrics used and the results obtained in comparison with well-known encryption methods such as 3DES and AES. Visual and statistical tests, encryption execution time, key size and key space, Normalized Mean Absolute Error (NMAE) and Peak Signal-to-Noise Ratio (PSNR), and Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are the metrics that will be presented here.

Palm (512×512) | Green Technology (1024×1024) | Secret Message (512×512)

Fig. 1. Selected images from experiments.

A. Visual and Statistical Tests

To prevent attackers from predicting the source image, high confusion, and diffusion effects must be achieved in the encoded image. This is one of the main objectives of any image encoding method. As depicted in Fig. 2, the suggested image encoding technique successfully produced encrypted images with a significant level of distortion.

Statistical histogram analysis of an image is a method used by attackers to gather information about the color/byte intensity distribution in the image. A high degree of flatness in the color/byte histogram of the encoded image provides immunity to statistical analysis attacks. Fig. 3 depicts the histograms of the source images in Fig. 1 and the encrypted images in Fig. 2.

Palm (512×512) | Green Technology (1024×1024) | Secret Message (512×512)

Fig. 2. Encrypted images generated by encrypting the source images in Fig. 1.

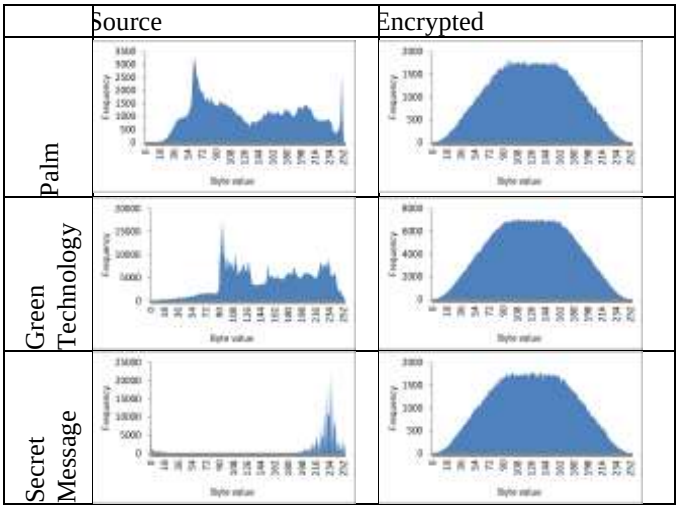

Fig. 3. The histograms of the source and encrypted images in Fig. 1 and Fig. 2 respectively.

DOI:http://doi.org/10.24086/cocos2024/paper.1509

B. Encryption Execution Time

The time it takes to encrypt data is a key factor when evaluating an encryption method. An efficient encryption method takes less time in the encryption process. The recorded times needed to complete the encryption process using the proposed image encryption method compared to known encryption methods (AES, 3DES) are listed in Table I. It is clear from Table I that the suggested encryption method achieves the lowest average encryption time.

TABLE I. RECORDED ENCRYPTION TIME IN THE SUGGESTED ENCRYPTION METHOD, AES, AND 3DES.

Image	**Encryption Time ET (sec)**		
	Suggested Method	***AES***	***3DES***
Palm	0.353	0.572	0.618
Green Technology	1.982	2.261	2.719
Secret Message	0.369	0.561	0.614
Average	0.901	1.131	1.317

C. Key Size and Key Space

The attackers usually use a brute-force attack to crack the key used in the encryption method. Using a random key and a large key length makes it difficult for attackers to guess the key and increases the time needed to crack the key. Whereas the known encryption methods (AES and 3DES) use key sizes of 256 and 128 bits respectively, the user of the proposed encryption method can use any digital file of any size to use it as a key. The proposed encryption method has a larger key size compared to the AES and 3DES methods. In addition to using a large key size, the frequent utilization of a secret RNGA in the encryption process makes it exceedingly challenging to decrypt the encoded image.

The key size (in bits) used in the suggested encryption method can be calculated using (4). Table II shows the key size and the key space for the AES, 3DES, and the proposed encryption methods [10], [11].

$$KeySize_{bit} = (Digital\ File\ Size_{byte}) \times 8 \quad (4)$$

TABLE II. KEY SIZE AND KEY SPACE FOR THE AES, 3DES, AND THE SUGGESTED ENCRYPTION METHODS.

Encryption Method	**Encryption key**	
	Key Size (bits)	***Key space***
AES	256	2^{256}
3DES	128	2^{128}
Suggested method	$KeySize_{bit}$	$2^{KeySize_{bit}}$

D. NMAE and PSNR Metrics

To perform numerical calculations of the amount of distortion achieved in encrypted images, Equations (5) and (6) are used to calculate the NMAE and PSNR metrics, respectively [12].

$$NMAE = \frac{\sum_{k=0}^{S_{Size}-1}|S(k)-E(k)|}{S_{Size}} \times 100 \quad (5)$$

$$PSNR_{db} = 10.\log_{10}\left(\frac{Max_S^2}{NMAE}\right) \quad (6)$$

Where S and E are the source and encrypted images, and Max_S is the maximum possible pixel value of S.

Tables III and IV show that the average values of the NMAE and PSNR indicate that the suggested encryption method succeeded in achieving competitive values compared to AES and 3DES methods.

TABLE III. NMAE VALUES FOR THE SUGGESTED ENCRYPTION METHOD, AES, AND 3DES.

Image	**NMAE (%)**		
	Suggested Method	***AES***	***3DES***
Palm	69.514	69.537	69.566
Green Technology	62.557	62.582	62.580
Secret Message	54.122	54.078	54.146
Average	62.064	62.066	62.097

TABLE IV. PSNR VALUES FOR THE SUGGESTED ENCRYPTION METHOD, AES, AND 3DES.

Image	**PSNR (%)**		
	Suggested Method	***AES***	***3DES***
Palm	6.488	6.485	6.481
Green Technology	6.713	6.713	6.712
Secret Message	5.616	5.622	5.612
Average	6.272	6.273	6.268

E. NPCR and UACI Metrics

Differential attacks are employed by hackers to assess the vulnerability of an encryption method against slight modifications in the source image. The efficiency of the encryption method is demonstrated by the ability to generate a completely different encrypted image by changing just a single pixel in the source image. To test this property, the NPCR and UACI values are calculated using (8) and (9), respectively.

$$D(i,j) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j) \\ 0, C_1(i,j) = C_2(i,j) \end{cases} \quad (7)$$

$$NPCR = \frac{\sum_{i=1}^{Width}\sum_{j=1}^{Height} D(i,j)}{Width \times Height} \times 100\% \quad (8)$$

$$UACI = \frac{\sum_{i=1}^{Width}\sum_{j=1}^{Height}|C_1(i,j)-C_2(i,j)|}{Width \times Height \times 255} \times 100\% \quad (9)$$

Tables V and VI show the NPCR and UACI values between the two encrypted images C_1 and C_2 that are produced using the proposed encryption method and other methods (AES and

DOI:http://doi.org/10.24086/cocos2024/paper.1509

3DES). From the values in Tables V and VI, it can be seen that the proposed encryption method successfully generated two different encrypted images when a single pixel of the source image changes.

TABLE V. NPCR VALUES BETWEEN THE TWO ENCRYPTED IMAGES C_1 AND C_2.

Image	**NPCR**		
	Suggested Method	***AES***	***3DES***
Palm	99.603	99.603	99.620
Green Technology	99.609	99.613	99.609
Secret Message	99.624	99.612	99.607
Average	99.612	99.609	99.612

TABLE VI. UACI VALUES BETWEEN THE TWO ENCRYPTED IMAGES C_1 AND C_2.

Image	**UACI**		
	Suggested Method	***AES***	***3DES***
Palm	33.491	33.443	33.434
Green Technology	33.483	33.462	33.491
Secret Message	33.445	33.465	33.471
Average	33.473	33.457	33.465

V. CONCLUSION

A matrix-based image encryption method has been introduced in this paper, after representing the source image data as a matrix and implementing a set of operations on the elements of the matrix to produce the encrypted image. Several factors contributed to the success of the suggested encryption method, such as the use of an external user key, and the adoption of RNGA to generate a high percentage of randomness in the sequence of execution of the selected matrix operations and the implementation of each of these operations. The comparative evaluation test showed that the proposed encryption method promises to be used efficiently in protecting images.

REFERENCES

[1] Abbas Fadhil Al-Husainy, M., & Al-Shargabi, B. (2020). Secure and lightweight encryption model for IoT surveillance camera. International Journal of Advanced Trends in Computer Science and Engineering, 9(2), 1840-1847. http://dx.doi.org/10.30534/ijatcse/2020/143922020

[2] Al-Haj, A., Abandah, G., & Hussein, N. (2015). Crypto-based algorithms for secured medical image transmission. IET Information Security, 9(6), 365-373. https://doi.org/10.1049/iet-ifs.2014.0245

[3] Seghier, A., Li, J., & Sun, D. Z. (2019). Advanced encryption standard based on key dependent S-Box cube. IET Information Security, 13(6), 552-558. https://doi.org/10.1049/iet-ifs.2018.5043

[4] Ye, G., & Huang, X. (2015). An image encryption algorithm based on autoblocking and electrocardiography. IEEE Multimedia, 23(2), 64-71. https://doi.ieeecomputersociety.org/10.1109/MMUL.2015.72

[5] Xian, Y., & Wang, X. (2021). Fractal sorting matrix and its application on chaotic image encryption. Information Sciences, 547, 1154-1169. https://doi.org/10.1016/j.ins.2020.09.055

[6] Wang, X., & Gao, S. (2020). Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. Information sciences, 507, 16-36. https://doi.org/10.1016/j.ins.2019.08.041

[7] Zhang, W., Yu, H., Zhao, Y. L., & Zhu, Z. L. (2016). Image encryption based on three-dimensional bit matrix permutation. Signal Processing, 118, 36-50. https://doi.org/10.1016/j.sigpro.2015.06.008

[8] Hu, X., Wei, L., Chen, W., Chen, Q., & Guo, Y. (2020). Color image encryption algorithm based on dynamic chaos and matrix convolution. IEEE access, 8, 12452-12466. https://doi.org/10.1109/ACCESS.2020.2965740

[9] Zhang, J., & Huo, D. (2019). Image encryption algorithm based on quantum chaotic map and DNA coding. Multimedia Tools and Applications, 78, 15605-15621. https://doi.org/10.1007/s11042-018-6973-6

[10] Tiessen, T., Knudsen, L. R., Kölbl, S., & Lauridsen, M. M. (2015). Security of the AES with a Secret S-Box. Lecture Notes in Computer Science, 175–189. https://doi.org/10.1007/978-3-662-48116-5_9

[11] Vaidehi, M., & Justus Rabi, B. (2015). Enhanced MixColumn Design for AES Encryption. Indian Journal of Science and Technology, 8(35). https://doi.org/10.17485/ijst/2015/v8i35/82302

[12] Mohammed Abbas Fadhil Al-Husainy, Hamza Abbass Al-Sewadi and Shadi R. Masadeh, "Using a DNA tape as a key for encrypt images", Int. J. Electronic Security and Digital Forensics, Vol. 14, No. 4, pp. 373-387, 2022.

DOI:http://doi.org/10.24086/cocos2024/paper.1509