An improvement for CAST-128 encryption based on magic square and matrix inversion

Suhad Muhajer Kareem[1], Ayad Al-Adhami[2], Abdul Monem S. Rahma[3]
[1]College of Computer Science and Information Technology, University of Basrah, Basrah, Iraq
[2]Department of Computer Science, University of Technology, Baghdad, Iraq
[3]Department of Computer Science, Al-Maarif University College, Anbar, Iraq

Article Info

Article history:

Received Nov 26, 2022
Revised Mar 24, 2023
Accepted Apr 17, 2023

Keywords:

Encryption
Histogram
Inverse matrix
Magic square
Matrix

ABSTRACT

This paper presents two image encryption methods which aim to improve the CAST 128-bit algorithm by increasing the security level of encrypted images. The first improvement uses a magic square of order three, while the second improvement uses a 2×2 matrix over GF(P). Both modifications are used in each round of the CAST algorithm, in place of a standard algorithm which uses XOR to increase the correlation between the plaintext and ciphertext. Simulations are carried out in order to evaluate the image encryption system with regard to complexity, time consumption, histogram, information entropy, differential attacks, noise evaluation, adjacent pixels' correlation index, National Institute of Standards and Technology (NIST) analyses, mean absolute error, and average difference. The experimental results demonstrate that the encryption and decryption time when using the proposed CAST 128-bit algorithm with magic square is less than the time required for the CAST 128-bit algorithm with the matrix. Conversely, the proposed CAST with the matrix is higher than the CAST with the magic square. Both theoretical analysis and experimental results confirm that the two proposed enhancements to CAST perform effectively with sufficient security levels.

Corresponding Author:

Suhad Muhajer Kareem
College of Computer Science and Information Technology, University of Basrah
Basrah, Iraq
Email: suhad.kareem@uobasrah.edu.iq

1. INTRODUCTION

With the rapid progress in communication technology, broadcasting images through networks has become routine. Digital image processing forms a large part of the data transmitted via the internet. However, illegal access to transmitted images is becoming a significant issue due to the capability of advanced computer processors. Therefore, securing images is essential for keeping the content of images confidential. Depending on the image application, image security against different types of attacks can be vital. Consequently, securing images involves information hiding, encryption, and steganography. Most encryption algorithms are used to protect sensitive information. The most notable example of image security is image encryption. Image encryption methods use several techniques, such as symmetric key encryption or public key encryption [1], [2]. In particular, image encryption methods are dependent on changes to pixel arrangement, called diffusion and confusion, which depend on changing the value of pixels. CAST 128 is a symmetric key encryption method which encrypts blocks of data, it was initiated by Carliste, Adams, and Stafford Tavared of Entrust Technology in 1996 [3]. CAST is based on sixteen rounds of a Feistel network that accepts key sizes ranging from 40 bits to 128 bits. It divides 64-bit plaintext into two sections: left-hand

32-bit and right-hand 32-bit. For the key scheduling process in CAST 128-bit, 16 pairs of sub-keys (Kmi and Kri) are computed from the key. Further, CAST 128 uses four primitive operations: addition (+) and subtraction (-) using modulo 232 arithmetic, bitwise ex-OR (^), and left circular rotation (<<<). CAST 128 uses the function F, which involves the use of four S-boxes, a left circular operation and four operations functions. Each S-box is of size 8×32, the round number is determined by the left circular rotation operation and four operation functions [4]–[6]. The overall work of CAST 128 [7]. A magic square is used in many encryption techniques in order to increase security. The magic square can be defined as a square integer matrix that has the same sum values in row, column, and main diagonal. In other words, if the magic square is of order four, the magic sum value is 24. The numbers in the rows, columns, and principal diagonals of a magic square of the third order (3×3) add up to a magic sum of 12, as illustrated in Figure 1 [8], [9].

1	6	5
8	4	0
3	2	7

Figure 1. Magic square of order 3

The majority of encryption algorithms use logic functions with two operators (0 and 1), such as XOR and ADD. The encryption methods used in these logic functions can be easily broken to retrieve the plaintext. Therefore, many researchers have proposed algorithms to overcome these problems. Such an algorithm would replace the XOR function with complex mathematical functions [10], [11]. This paper presents two modified proposals which are based on the CAST 128 algorithm and replace the XOR on the left side of Feistel CAST 128, the first is replaced by a magic square of order 3, while the second is replaced by an inverse matrix 3×3. These two proposals aim to enhance the mathematical operations in the CAST 128 algorithm, thus increasing the encryption algorithm's security level. The remainder of the paper is structured as: section 2 presents some of the relevant literature reviews, section 3 explains the proposed modifications to the CAST 128-bit algorithm, and finally sections 4 and 5 present security analysis and conclusions, respectively.

2. RELATED WORK

There are numerous cryptography techniques in digital image encryption, such as symmetric cryptography, chaotic theory, and quantum cryptography. Image encryption techniques depend on two significant steps: the first alters pixel arrangements, while the second alters the values of the pixels. This section presents some of the most relevant related works on modifications to encryption algorithms using magic squares and inverse arrays that can be applied to digital images. Encryption algorithm proposed to improve the mix column and shift row in the original advanced encryption standard (AES) representing S-boxes, applying as S-boxes, and apply multiplication process inside the encryption [12]. While the inverse matrix is used inside the decryption, a modified AES is used for encrypting audio wave files. The applicability of evaluating a magic square matrix of any order for encryption and decryption is considered [13], [14]. However, the inclusion of 2.13 (factors of 26) has been the most significant limitation for the implementation of hill and magic square ciphers in cryptographically investigations, especially in the decryption process. The effectiveness of a cryptographic method is determined by the required time for encryption/decryption and the manner in which it generates distinct ciphertexts from plaintext. Weak magic squares (for even numbers, n) provide ciphertext that is maximally distinct from plaintext, as opposed to true magic squares. Encryption/decryption of any matrix of magic squares is made possible by the introduction of dummy letters in addition to the existing 26 letters in the english alphabet. The incorporation of selected dummy letters not only simplifies the encryption/decryption procedure but also eliminates duplicate letters (vowels) within a message. The encryption/decryption procedure has been improved and provides an additional layer of security to any public-key cryptosystem employing a magic square or a poor implementation of a magic square.

The same is true for [15] where the proposed a development that combines symmetric cryptographic models, asymmetric cryptographic models, and a magic cube. In the construction phase, Diffie-Hellman was used to define the magic square category as odd, singly-even or doubly-even, both the starting number, and the difference of the value were determined. The Diffie-Hellman algorithm was utilized to determine the dimension of the magic cube's construction, the type of magic square (odd, singly-even, or doubly-even), the starting number and the difference value, in addition to the face or dimension number used to generate the ciphering key for both parties. Thiagarajan *et al.* [16] developed a method of encrypting and decrypting a message using a key and a cyclic square matrix. This method can be used for any number of words with the most characters and longest word. The researchers additionally discussed how long the algorithm would take to run. The proposed algorithm is simple but would be difficult to break. Rahma and Abbas [17] explained an enhancement of the AES algorithm using different sizes of data matrices created by multiplying irreducible polynomials of different orders (2.4 and 8). In testing, these modifications provided effective security, as the use of multiple irreducible polynomials of different orders of degree increased the randomness efficiency and speed.

Mohammed and Hasan [18] proposed method thats planned alterations to the operation of data encryption standard (DES) to ensure a high level of security. Among these modifications, is the implementation of matrix multiplication in place of the XOR operator. Moreover, each round employs four keys, two of which are obtained from the primary key and two of which are produced internally. The four keys are used in a specific order using round numbers. The primary key is derived from a 64-byte random string. This key is then enlarged and divided across 16 keys. Based on the mathematical theory of GF(2), the suggested algorithm improves upon the DES algorithm. Rather than the XOR technique, matrix multiplication is employed, which enables substitution and permutation of each multiplication process. The XOR technique also assists in reducing the time required for encryption and decryption. This method's effectiveness relies on the utilization of two diagonal key matrices created from the randomly generated primary key. The operations of the two dynamic internal keys, the matrix multiplication and the replacement of the previous XOR make it difficult to attack the known plaintext. Alattar and Rahma [19] considered developing encryption methods using an order 5 magic square. Both GF (P) and GF (2^8) were used to encrypt text and images. With an arbitrary number of rounds added, the two modified algorithms were utilized, the first with a message length of 10 and the second with a message length of 14. The text from the first round served as the input text for the subsequent round through the use of a mask, which multiplied in the odd rounds and added in the even rounds.

3. METHOD

CAST is a problematic block encryption algorithm as it uses functions that are based on two binary operations (0 and 1) and XOR. In order to enhance the security of the algorithm against attacks, this paper proposes two new modifications to CAST 128. This is achieved by substituting the XOR function within the sixteen rounds of Feistel in the CAST 128-bit algorithm. The proposed alterations will be explained in the following sections. The CAST 128 method, a secret-key block cipher, has been modified in this work in an effort to improve performance. The modified CAST 128 algorithm is designed to increase security and decrease overall data encryption and decryption time. The goal of this study is to increase performance while maintaining the existing CAST 128 algorithm's security, simplicity and memory requirements. The CAST 128 Feistel network's function F is the sole aspect of the proposed modification that has been altered. The very high-speed integration circuit hardware description language (VHDL) application was created to demonstrate the variations in the delay, as the change in the overall time required for encryption and decryption cannot be observed in software implementation.

3.1. First proposal: CAST 128 with magic square

This variation introduces a modification to the CAST 128-bit algorithm which aims to solve such issues by employing a magic square of order three in place of the XOR. This is achieved by substituting the ordinary XOR operation between the right side, after applying the function, and the left side in each round. When using a magic square for encryption, several steps must be followed: first, the magic square is created by utilizing magic square of order 3. Second, the mask is multiplied using the magic square. Finally, the summarisation of the specific row, column, and diagonal of the last matrix of order three is computed in the ciphertext, as shown in algorithm [8] which uses (1) to (6) for apply the summarisation.

$$Key_1 + Key_2 + pl_1 = sum_1 \tag{1}$$

$$Pl_4 + Key_3 + pl_3 = sum_2 \tag{2}$$

$$Key_1 + Pl_2 + Pl_4 = sum_3 \tag{3}$$

$$Pl_1 + Pl_3 + Pl_6 = sum_4 \tag{4}$$

$$Key_1 + Key_3 + Pl_6 = sum_5 \tag{5}$$

$$Pl_1 + Key_3 + Pl_4 = sum_6 \tag{6}$$

On the decryption side, the recipient has utilized the following steps to supplement the decryption procedure: first, an augmented matrix (AA) is constructed using the linear equation system of magic square of order 3, the ciphertext is swaped with the last column of the matrix (AA) by subtracting the last known value of the key, then reducing the matrix (AA) by deleting 1, 2, and 5 columns. The magic square of order 3 is regenerated by replacing the plaintext with the result of the Gaussian elimination using the last known value of the key. To determine the number of rounds, the magic square is multiplied by the inverse of the encryption mask to gain the resulting algorithm [8]. The modified version of the suggested CAST algorithm is displayed for each round in Figure 2 and Algorithm 1.

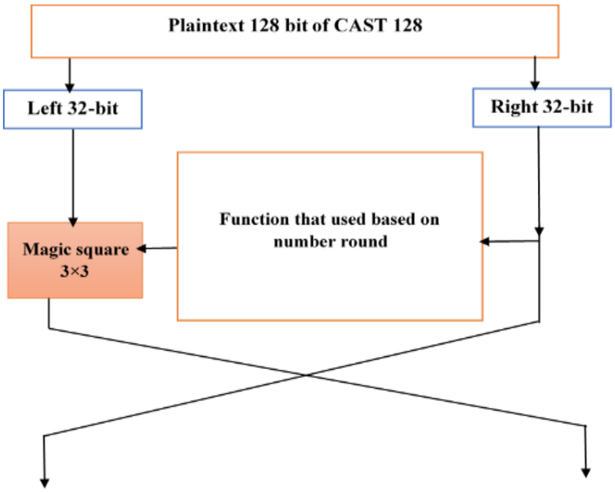

Plaintext 128 bit of CAST 128
Left 32-bit
Right 32-bit
Function that used based on number round
Magic square 3×3

Figure 2. Proposed CAST 128-bit with magic square of order 3

```
Algorithm 1: CAST 128 with magic square
Input:  plaintext $pl_1$...$pl_{64}$; key = $key_1 ... key_{128}$.
Output: Ciphertext $Ci_1 ... Ci_{64}$.
  Step1. Key schedule: Calculate sixteen pairs of subkeys $\{Key_{Pli}, Key_{ri}\}$ from Key.
  Step2. Divided 64 plaintext into left (L) 32-bit and right (R) 32-bit halves.
  Step3. Li and Ri can be calculated, where Li represents elements in matrix's left and Ri represents elements in matrix's right, as follows, for sixteen rounds:
    Li = Ri-1;
    Ri = apply magic square of order 3 ( L i-1 f(R i-1,$Key_{mi}, Key_{ri}$)), where f is of Type 1, Type 2, or Type 3, depending on i.
      End for
  Step4. (R16, L16 → (c1...c64.
  Step5. Replace the final L16, R16 blocks and then combine them to find the cipher text. However, the rounds (and subsequently the sub-key pairs) are used to compute (L00, R00).
```

3.1.1. Example

Input: plaintext=5-5-5-5-5 and 5, output: ciphertext, selected key=1, 2, and 3. The magic square of order 3 will be filled with plaintext and key as:

Key	Key	Pl
Pl	Key	Pl
Pl	Pl	Pl

1	2	5
5	3	5
5	5	5

Next, a mask is created to apply the encryption and decryption processes.

- The encryption processes: after applying the multiplication operations between the magic square and the mask, calculating the six sums defined in (1) to (6) will serve as a visual representation of the encryption process: Sum 1=11, Sum 2=15, Sum 3=8, Sum 4=15, Sum 5=9, Sum 6=13. The value of the ciphertext supplied to the opposite side is represented by these sums.

- The decryption processes: the recipient will receive the encrypted text as well as the key for the decryption operation used to obtain the plaintext. The six unknowns will be imposed by X1–X6, where X1–X6 are new ciphers for decryption and the inverse mask will be used to obtain the plaintext.

1	2	X1
X2	3	X3
X4	X5	X6

3.2. Second proposal: CAST 128 with matrix and inverse

The second method proposed in this paper replaces the XOR operation in each round of the original CAST 128-bit Feistel algorithm with a 2×2 matrix between the left and right sides (after applying the function). The modification on each round of the proposed algorithm is shown in Algorithm 2 and Figure 3. This modification is used for enhancing security level by using uses matrix inversion process instead of using normal XoR bit processing.

```
Algorithm 2: CAST 128 with matrix
Input: plaintext m11...m664; key= key1...key128.
Output: cipher text c1...c64.
Step1. Key-schedule: Calculate sixteen pairs of sub keys $\{Key_{mi}, Key_{ri}\}$ from Key.
Step2. Divided 64 plaintext into 32-bit segments on the left (L) and right (R).
Step3. Li and Ri can be calculated, where Li represents elements in matrix's left and Ri represents elements in matrix's right, as follows, for sixteen rounds:
    Lii = Rii-1;
    Rii = apply matrix 2× 2 (Lii-1 f (Rii-1, $Key_{mi}, Key_{ri}$)), where f is of Type 11, Type 22, or Type 33, depending on i.
End for
Step4. (R16, L16)→(c1...c64).
Step5. To create the encrypted text, replace the final L16 and R16 blocks, then merge them.
```

The encryption procedure described above is utilised for both encryption and decryption, however the rounds (and therefore the sub keys pairs) are employed in reverse order to computer (L00, R00) from (R116, L116).

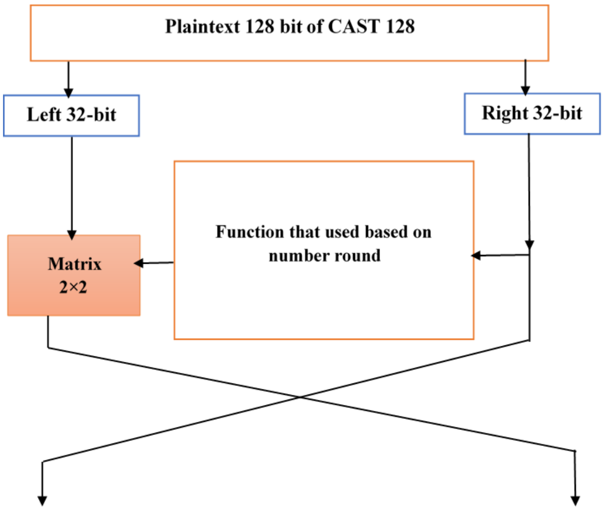

Plaintext 128 bit of CAST 128
Left 32-bit
Right 32-bit
Function that used based on number round
Matrix 2×2

Figure 3. Proposed CAST 128-bit with matrix 2×2

3.2.1. Example compute matrix with mode prime 251 in encryption side

The folowing example clarifietes how apply multiplication operation between 32-bit of plaintext and key based on matrix with mode prime 252 in encryption side of CAST 128 algorithm.

$Pl_{0,0}$	$Pl_{0,1}$
$Pl_{1,0}$	$Pl_{1,1}$

$Key_{0,0}$	$Key_{0,1}$
$Key_{1,0}$	$Key_{1,1}$

Result Pl×Key=	($Pl_{0,0}$×$Key_{0,0}$+$Pl_{0,0}$× $Key_{0,1}$) mod prime number	($Pl_{0,0}$× $Key_{0,0}$+$Pl_{0,0}$×$Key_{0,1}$) mod prime number
	($Pl_{1,0}$×$Key_{1,0}$+$Pl_{1,1}$ × $Key_{1,1}$) mod prime number	($Pl_{0,0}$× $Key_{0,0}$+$P_{0,0}$×$K_{0,1}$) mod prime number

where inverse matrix used in decryption side

$$\begin{bmatrix} 7 & 7 \\ 3 & 29 \end{bmatrix}^{-1} = \begin{bmatrix} 156 & 222 \\ 131 & 29 \end{bmatrix}$$

4. SECURITY METRICS

This section presents the results of the proposed algorithms. The proposed algorithms are implemented using C# and MATLAB. Five PNG and JPEG colour images were employed. To evaluate the performance of the proposed methods, several security tests based on the original and encrypted image were performed.

4.1. Complexity

Encryption algorithms are uncover to numerous types of attacks, like brute force attacks in which the assailant attempts all probale keys to break the algorithm and retrieve the original text. As such, the layout of the proposed algorithm rises resilience versus this type of attack by increasing the complexity of the algorithm. The complexity of the original CAST 128-bit algorithm is computed as: $2\times(2)^8\times32\times2=2\times(2)^8\times2^5\times2=2^{15}$. The security of the CAST 128-bit algorithm with the magic square is calculated using the additional key and magic square: $2\times(2)^8\times48\times2\times(256)^5\times(2)^8\times3\times9=2^{27}\times(256)^5\times3\times9$. Finally, the security of the CAST 128-bit with matrix algorithm is calculated based on employing matrix 2×2: $(2^1)^{128}\times(2^2)^{64}\times(2^4)^{32}\times(2^1)^{16}\times32\times2\times2\times2$.

4.2. Time consuming

One of the most important security indicators for any encryption system's precision is encryption time. The accuracy of encryption is inversely correlated with encryption time, meaning that the more quickly an algorithm can encrypt data, the more effective it will be. The CAST with magic square outperforms other algorithms when comparing the encryption times required by CAST and CAST with matrix. Results are obtained by comparing the two proposed algorithms using five colour images as shown in Table 1.

Table 1. Time execution for two proposed

Image name	CAST with magic square of order 3		CAST with magic matric 2×2	
	Time for enc.	Time for dec.	Time for enc.	Time for dec.
Lena	12	26	24	48
Cat	36	29	52	45
Airplane	16	56	40	62
Fruit	9	16	28	49
Peppers	6	15	19	37

4.3. Histogram metric

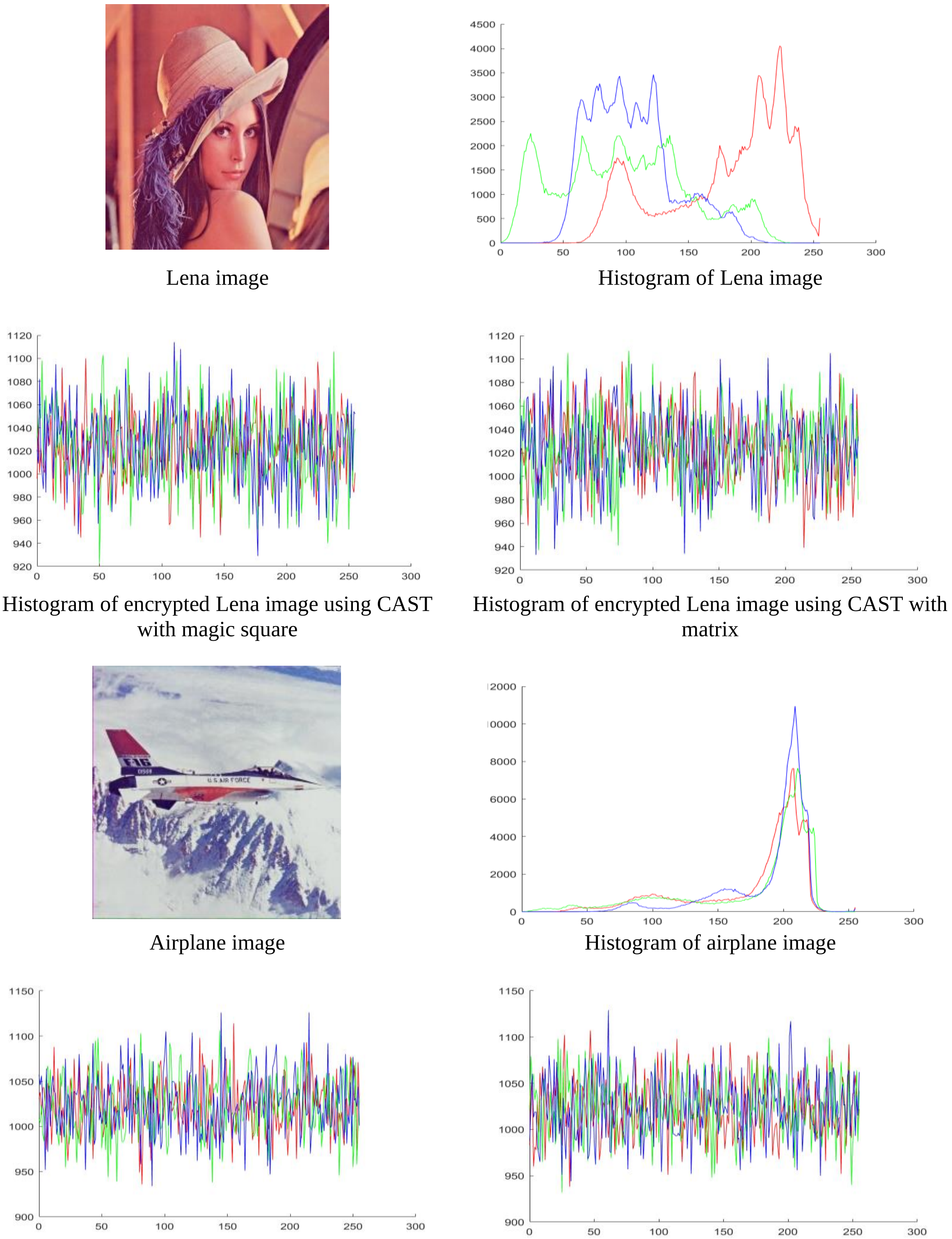
Another useful statistical test is a histogram, which is used to show the distribution of pixels in images that occur at various intensity values. A strongly encrypted image must have a uniform histogram to resist any statistical cryptanalysis [20], [21]. Figure 4 displays the histograms of both original images and encrypted images. The histogram of the ciphered image differs in all components (red, green, and blue) from the plain image. Figure 4 shows the proposed algorithms get good results where the encrypted images using two proposed encryption methods (Lena and airplane), show uniform distribution of image points.

4.4. Correlation coefficient analysis

The correlation coefficient is used to measure the encryption quality of the proposed algorithms. The correlation coefficient is 1 when images are highly connected and the correlation coefficient is almost 0 when images are encrypted. In the horizontal, vertical, and diagonal axes, there is typically a high correlation between the neighbouring pixels of the original image. The correlation coefficients of the pixels in the encrypted image can be made to have a sufficiently low correlation using an ideal encryption technique to defend against statistical attacks. The correlation coefficient can be computed using (7) [22]–[24]:

$$C = \frac{E[(x-M_x)(y-M_y)]}{S_x S_y} \tag{7}$$

In which M_x and M_y represent the mean of x and y respectively and $E(.)$ represents expectancy. The standard deviations of x and y are denoted by S_x and S_y respectively. Table 2 presents the values of the correlation coefficient for the encrypted images generated by the two proposed modifications in three directions: horizontal and vertical.



Lena image

Histogram of Lena image

Histogram of encrypted Lena image using CAST with magic square

Histogram of encrypted Lena image using CAST with matrix

Airplane image

Histogram of airplane image

Histogram of encrypted airplane image using CAST with magic square

Histogram of encrypted airplane image using CAST with matrix

Figure 4. Histogram distribution of original and encrypted from Lena and airplane

Table 2. Correlation coefficient for two proposed

	Correlation coefficient for:			
	CAST-128 bit with magic square		CAST 128-bit magic matric 2×2	
Image name	The vertical correlation coefficient	The horizontal correlation coefficient	The vertical correlation coefficient	The horizontal correlation coefficient
Lena	0.4072	0.0364	-0.0068258	0.0038556
Cat	0.00015	0.002273	-0.0026	-0.00088
Airplane	0.4072	0.03643	0.00177	-0.000978
Fruit	0.87098	0.02903	-0.00082	0.0031603
Peppers	0.00065	0.008847	-0.00499	-0.00142

4.5. Entropy analyses

Entropy testing is a statistical method used to determine whether the encrypted image is random. Additionally, it demonstrates the amount of information that a ciphered image contains. The value of the entropy for the cipher image C [25], [26] can be computed using (8):

$$H(C) = \sum_{i=0}^{2^N-1} P(c_i)\log_2 \frac{1}{P(c_i)} \tag{8}$$

In (8), $P(c_i)$ represents the probability of occurrence of the symbol c_i in cipher image C. The value of the entropy of encrypted image C and the ideal value of entropy H(C) is 8. In our two proposed methods, the entropy values calculated for encrypted image C are close to the ideal value as shown in Table 3.

Table 3. Entropy analyses for two proposed

		Entropy analyses for:	
Image name	Original image	CAST-128 bit with magic square	CAST-128 bit with magic matric 2×2
Lena	7.4414	7.7337	7.6329
Cat	3.4444	7.6917	7.5222
Airplane	6.7167	7.8830	7.78332
Fruit	7.0563	7.6587	7.6228
Peppers	7.5797	7.6311	7.6316

4.6. Analysis of differential attack

Two measures that can be used to test the performance of cryptography systems are the number of pixels of change rate (NPCR) and unified average changing intensity (UACI). NPCR is used to assess the impact on the encrypted image of modifying a single pixel of the plain image. The average difference in colour intensity between two photos is calculated using UACI [27]–[29]. The results of these two measures are displayed in Table 4.

$$NPCR = \frac{\sum_{ij} D(i,j)}{W \times H} \times 100\% \tag{9}$$

where $D(i,j) = \begin{cases} 0, C1(i,j) \neq C2(i,j) \\ 1, C1(i,j) = C2(i,j) \end{cases}$.

$$UACI = \frac{1}{W \times H} \sum_{i,j} \frac{C1(i,j) - C2(i,j)}{255} \times 100\% \tag{10}$$

Where H, W represent the height and width of the image.

Table 4. NPCR and UACI for two proposed

	Resistance to differential attack results of the NPCR and UACI measure for:			
Image name	CAST-128 bit with magic square		CAST-128 bit with magic matric 2×2	
	NPCR	UACI	NPCR	UACI
Lena	99.61	30.42	99.60	30.42
Cat	99.61	44.55	99.60	44.09
Airplane	99.62	32.65	99.62	32.62
Fruit	99.62	30.51	99.60	32.63
Peppers	99.61	32.13	99.61	32.14

4.7. Peak signal to noise ratio

Peak signal to noise ratio (PSNR), is used to calculate the comparison between the original image and the encrypted image. The higher the PSNR, the closer the encrypted image is to the original, which correlates to a higher-quality image [30]. The two proposed methods produce convergent results for PSNR as shown in Table 5. Table 5 presents the results of the PSNR comparison between the original image, the image encrypted by CAST 128-bit with magic square, and the image encrypted by CAST 128-bit with matrix. These results demonstrate that the two proposed methods generate higher PSNR values than the original image. Therefore, the encrypted images are of higher quality.

Table 5. PSNR for two proposed

	Encryption quality for:			
Image name	CAST-128 bit with magic square		CAST-128 bit with magic matric 2×2	
	PSNR	MSE	PSNR	MSE
Lena	28.691	169.81	28.694	169.93
Cat	25.095	209.34	25.378	206.19
Airplane	25.950	167.07	25.999	164.64
Fruit	27.141	122.12	27.296	136.33
Peppers	27.927	56.83	26.727	57.09

4.8. Comparison results

Several protocols on CAST algorithm were proposed. As mentioned in section 2, however these protocol has some limitations regarding different security metrics. Table 6 summarizes the result.

Table 6. Comparison results

	Randomness	PSNR	Entropy	NPCR and UACI	Histogram metric
The two proposed protocols	✓	✓	✓	✓	✓
[16]	✓	✗	✗	✗	✗
[17]	✓	✗	✗	✗	✗
[18]	✓	✗	✓	✗	✗
[19]	✓	✗	✓	✗	✓
[20]	✓	✗	✓	✓	✓

5. CONCLUSION

Two modifications of the CAST 128-bit encryption algorithm have been proposed, using a magic square and a matrix. In the CAST 128-bit with magic square, a magic square of order 3 is used in place of an ordinary XOR process in every tour of the Feistel algorithm, a randomly generated key is used to fill the magic square. In the CAST 128-bit with matrix, a 2×2 matrix is used within each round of the algorithm in place of the XOR operation. A simultaneous result is presented on different coloured images in order to compare the two proposed CAST 128-bit algorithms. The proposed scheme is applied to the simulation framework and evaluates the security and performance using several measurements such as complexity, time to execute, histogram, correlation coefficient, entropy, key space, NPCR, UACI, PSNR, and National Institute of Standards and Technology (NIST) test in order to ensure the proposed algorithm is resistant to statistical attacks and brute force attack. The proposed methods perform well in many security tests. The proposed CAST with magic square requires less time to execute than the proposed matrix-based method. Conversely, the modification on CAST using a matrix is of high complexity compared with the CAST with magic square method. The proposed methods have been evaluated against noise, statistical, and differential attacks.

REFERENCES

[1] M. N. A. Wahid, A. Ali, B. Esparham, and M. Marwan, “A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention,” *Journal of Computer Science Applications and Information Technology*, vol. 3, no. 2, pp. 1–7, 2018.

[2] X. Zhang, L. Wang, Y. Niu, G. Cui, and S. Geng, “Image Encryption Algorithm Based on the H-Fractal and Dynamic Self-Invertible Matrix,” *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–12, 2019, doi: 10.1155/2019/9524080.

[3] K. D. Muthavhine and M. Sumbwanyambe, “Modifying CAST algorithm in order to increase encryption strength and to reduce memory limitations,” in *2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 2021, pp. 1–7, doi: 10.1109/icABCD51485.2021.9519349.

[4] M. Arora, S. Sharma, and D. Engles, “Parametric comparison of EMDS algorithm with some symmetric cryptosystems,” *Egyptian Informatics Journal*, vol. 18, no. 2, pp. 141–149, 2017, doi: 10.1016/j.eij.2016.11.004.

[5] L. E. Tariq and E. Falih, “Image encryption and decryption using CAST-128 with proposed adaptive key,” *Journal of College Oeducation*, vol. 1, no. 5, pp. 89–100, 2019.

[6] Z. Hercigonja, “Comparative Analysis of Cryptographic Algorithms,” *International Journal of Digital Technology & Economy*, vol. 1, no. 2, pp. 127–134, 2016.

[7] C. Rekha and G. N. Krishnamurthy, “An optimized key scheduling algorithm for CAST-128 using dynamic substitution S-box,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 2585–2590, 2019, doi: 10.35940/ijrte.C4920.098319.

[8] R. H. A. -Hashemy and S. A. Mehdi, “A new algorithm based on magic square and a novel chaotic system for image encryption,” *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1202–1215, 2020, doi: 10.1515/jisys-2018-0404.

[9] P. Bartlett, “Latin Squares and Magic,” *Mathcamp*, pp. 1–6, 2012.

[10] H. B. A. Wahab, A. M. S. Rahma, and H. M. Y. A. -Bayatti, “Proposed New Quantum Cryptography System Using Quantum Description Techniques for Generated Curves.,” in *International conference on Security and Management*, 2009, pp. 658–664.

[11] M. Obaidat, J. Brown, S. Obeidat, and M. Rawashdeh, “A hybrid dynamic encryption scheme for multi-factor verification: A novel paradigm for remote authentication,” *Sensors*, vol. 20, no. 15, pp. 1–32, 2020, doi: 10.3390/s20154212.

[12] N. H. M. Ali, “An Improved AES Encryption of Audio Wave Files,” University of Technology, 2015.

[13] K. G.N, R. V, L. G.H, and A. M.E, “Performance enhancement of CAST-128 algorithm by modifying its function,” in *Advances in Computer and Information Sciences and Engineering*, Dordrecht: Springer, 2008, pp. 256–260, doi: 10.1007/978-1-4020-8741-7_46.

[14] I. Tomba and N. Shibiraj, “Successful Implementation of the Hill and Magic Square Ciphers: A New Direction,” *International Journal of Advanced Computer Technology (IJACT)*, vol. 2, no. 3, pp. 83–91, 2013.

[15] O. A. Dawood, A. M. S. Rahma, and A. M. J. A. Hossen, “New Variant of Public Key Based on Diffie-Hellman with Magic Cube of Six-Dimensions,” *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 13, no. 10, pp. 31–47, 2015.

[16] K. Thiagarajan, P. Balasubramanian, J. Nagaraj, and J. Padmashree, “Encryption and decryption algorithm using algebraic matrix approach,” *Journal of Physics: Conference Series*, vol. 1000, no. 1, pp. 1–7, 2018, doi: 10.1088/1742-6596/1000/1/012148.

[17] A. M. Rahma and A. Abbas, “A Modified Matrices Approach in Advanced Encryption Standard Algorithm,” *Engineering and Technology Journal*, vol. 37, no. 3, pp. 86–91, 2019, doi: 10.30684/etj.37.3b.4.

[18] S. D. Mohammed and T. M. Hasan, “Cryptosystems using an improving hiding technique based on latin square and magic square,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 510–520, 2020, doi: 10.11591/ijeecs.v20.i1.pp510-520.

[19] I. M. Alattar and A. M. S. Rahma, “A new block cipher algorithm that adopts the magic square of the fifth order with messages of different lengths and multi-function in GF(28),” *Periodicals of Engineering and Natural Sciences*, vol. 9, no. 3, pp. 568–578, 2021, doi: 10.21533/pen.v9i3.2205.

[20] A. K. Shibeeb, M. H. Ahmed, and A. H. Mohammed, “A new chaotic image cryptosystem based on plaintext-associated mechanism and integrated confusion-diffusion operation,” *Karbala International Journal of Modern Science*, vol. 7, no. 3, pp. 176–188, 2021, doi: 10.33640/2405-609X.3117.

[21] M. Ahmad, E. A. Solami, X. Y. Wang, M. N. Doja, M. M. S. Beg, and A. A. Alzaidi, “Cryptanalysis of an image encryption algorithm based on combined chaos for a BAN system, and improved scheme using SHA-512 and hyperchaos,” *Symmetry*, vol. 10, no. 7, pp. 1–18, 2018, doi: 10.3390/sym10070266.

[22] C. Xu, J. Sun, and C. Wang, “A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems,” *Multimedia Tools and Applications*, vol. 79, no. 9–10, pp. 5573–5593, 2020, doi: 10.1007/s11042-019-08273-x.

[23] J. Pappachan and J. Baby, “Tinkerbell Maps based Image Encryption using Magic Square,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 4, no. 7, pp. 6226–6232, 2015, doi: 10.15662/ijareeie.2015.0407034.

[24] Y. Alghamdi, A. Munir, and J. Ahmad, “A lightweight image encryption algorithm based on chaotic map and random substitution,” *Entropy*, vol. 24, no. 10, pp. 1–25, 2022, doi: 10.3390/e24101344.

[25] Y. Wu, J. P. Noonan, and S. Agaian, “Shannon Entropy based Randomness Measurement and Test for Image Encryption,” *Information Sciences*, vol. 1, pp. 1–23, 2018.

[26] G. Liu, W. Li, X. Fan, Z. Li, Y. Wang, and H. Ma, “An Image Encryption Algorithm Based on Discrete-Time Alternating Quantum Walk and Advanced Encryption Standard,” *Entropy*, vol. 24, no. 5, pp. 1–16, 2022, doi: 10.3390/e24050608.

[27] J. Chandrasekaran and S. J. Thiruvengadam, “A Hybrid Chaotic and Number Theoretic Approach for Securing DICOM Images,” *Security and Communication Networks*, vol. 2017, pp. 1–12, 2017, doi: 10.1155/2017/6729896.

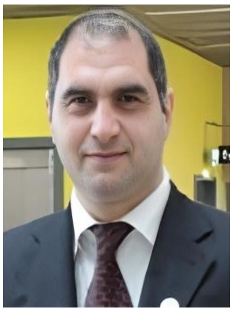
[28] S. Jassim and S. M. Hameed, “A Modified Advanced Encryption Standard for Color Images,” *Iraqi Journal of Science*, vol. 63, no. 1, pp. 294–312, 2022, doi: 10.24996/ijs.2022.63.1.29.

[29] G. Shraida and H. Younis, “An Efficient Diffusion Approach for Chaos-Based Image Encryption and DNA Sequences,” *Iraqi Journal for Electrical and Electronic Engineering*, vol. 18, no. 2, pp. 69–74, 2022, doi: 10.37917/ijeee.18.2.9.

[30] X. Song, M. Shi, Y. Zhou, and E. Wang, “An Image Compression Encryption Algorithm Based on Chaos and ZUC Stream Cipher,” *Entropy*, vol. 24, no. 5, pp. 1–21, 2022, doi: 10.3390/e24050742.

BIOGRAPHIES OF AUTHORS

Suhad Muhajer Kareem is lecture at College of Computer Science and Information Technology, University of Basrah, Iraq. She holds a Ph.D. degree in computer science with data security. Her research areas are image/signal processing, security, data mining, and text mining. She can be contacted at email: suhad.kareem@uobasrah.edu.iq.



Ayad Al-Adhami is a lecturer in computer security and his role in the Department of Computer Science is as the head of the computer security branch. He earned his Ph.D. degree in computing–computer security from Plymouth University, Plymouth, United Kingdom, 2018. His M.Sc. degree was in mathematics and computer applications, specified in information security, from AlNahrain University Baghdad, Iraq, 2008. His research interests include computer security, RFID security, and cryptography. He can be contacted at email: Ayad.h.ibrahim@uotechnology.edu.iq.

Abdul Monem S. Rahma have an extensive background in the field of cryptography and information security. In 1984, he received his Ph.D. in computer science from the Loughborough University of Technology in the United Kingdom and become a professor in computer science since 2008. He was the deputy dean of the Department of Computer Science, University of Technology, Baghdad, Iraq from 2005 to 2013, then from 2013 to 2015 become the dean of the the same department. Prof. Rahma is currently the head of the Department of Computer Science, Al-Maarif University College, Anbar, Iraq. He can be contacted at email: monem.rahma@uoa.edu.iq.