Indonesian Journal of Electrical Engineering and Computer Science
Vol. 20, No. 1, October 2020, pp. 510~520
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v20.i1.pp510-520

Cryptosystems using an improving hiding technique based on latin square and magic square

Sahab Dheyaa Mohammed[1], Taha Mohammed Hasan[2]
[1]University of Information Technology and Communications, Iraq
[2]University of Diyala, College of Science, Iraq

Article Info

Article history:

Received Feb 5, 2020
Revised Apr 7, 2020
Accepted Apr 19, 2020

Keywords:

Affine cipher function
Encryption process
Finite fields
Irreducible polynomial
Latin square

ABSTRACT

Hackers should be prevented from disclosing sensitive data when sent from one device to another over the network. Therefore, the proposed method was established to prevent the attackers from exploiting the vulnerabilities of the redundancy in the ciphertext and enhances the substitution and permutation operations of the encryption process the solution was performed by eliminates these duplicates by hiding the ciphertext into a submatrix 4x4 that chooses randomly from magic square 16x16 in each ciphering process. Two techniques of encrypted and hiding were executed in the encryption stage by using a magic square size 3×3 and Latin square size 3x3 to providing more permutation and also to ensure an inverse matrix of decryption operation be available. In the hiding stage, the ciphertext was hidden into a 16×16 matrix that includes 16 sub-magic squares to eliminate the duplicates in the ciphertext. Where, all elements that uses were polynomial numbers of a finite field of degree Galois Fields GF (2^8). The proposed technique is robust against disclosing the repetition encrypted data based on the result of Avalanche Effect in an accepted ratio (62%) and the results of the output of the proposed encryption method have acceptable randomness based on the results of the p-values (0.629515) of the National Institute of Standards and Technology (NIST) randomness tests. The work can be considered significant in the field of encrypting databases because the repetition of encrypted data inside databases is considered an important vulnerability that helps to guess the plaintext from the encrypted text.

Corresponding Author:

Sahab Dheyaa Mohammed,
University of Information Technology and Communications,
Baghdad, Iraq.
Email: sahab7dia@gmail.com

1. INTRODUCTION

Encryption that uses a peculiar number system is a good method to encode and decode data and provides additional security against attacks during transmission and providing opportunities for full encryption and decryption whilst hiding all technical details [1]. Numerous techniques are used to secure file transfer, including the types of encryption techniques designed to keep files secure. Substitution and transposition are two mechanisms used in symmetric encryption. Substitution involves changing plain text values to cipher text values. By contrast, the transposition moves the locations of plain text values [2].

Nowadays, attackers are trying to break the encryption algorithm by retrieving the key, or by analyzing a collision or the existence of repeated bits / characters (bytes) in the encrypted message to gain the algorithm of encryption or the key utilized for it. Therefore, the encryption method must be efficient and exclude repeated terms and the attacker cannot track the repetition [3]. The traditional cryptographic algorithm suffers from the problem of data redundancy in the ciphertext. This proposed method aims to develop an algorithm to exclude redundancy in the ciphertext. Where, new encryption and hiding algorithms

are implemented, using magic square and Latin square to increase the permutation and substitution to make encryption more secure and complex. In multiple encryption methods, many mathematical models and operations are used to improving encryption methods such as matrix multiplication, magic square, and Latin square. In this paper, we used the magic square and Latin square to improve the proposed encryption method and hide encrypted data.

1.1. Magic square and latin square

A magic square is a square matrix of integers with the same sum of the values in the rows, columns and main diagonals. As shown in Figure 1, a magic square of the fourth-order (i.e., 4×4) has a magic sum of 24, which is the total sum of the values in the rows, columns and main diagonals. A total of 880 different fourth-order magic squares are provided [4]. A Latin square is an n×n array of order n, in which all rows and columns contain $\{0, 1, 2 \ ... \ n - 1\}$ precisely once and the also the symbols occur precisely once in each row and column. A Latin square is called diagonalise.

Latin square when the square has a main diagonal in a transversal form. Latin squares can be constructed using number theory using only one step from the magic square, particularly by using the modulo n of the magic square. In this manner, the Latin square obtained are of two types, namely, diagonalised and doubly diagonalised Latin squares of any odd order n [5]. That is, a magic square is an n×n array with integers $\{0; \ 1 \ ... \ n^2 - 1\}$, such that, each number is filled once in each row and column and the sum of each row, column and main diagonal or main antidiagonal is the same constant value. An example for order 3 as in Figure 2 [6].

1	2	16	15
13	14	4	3
12	7	9	6
8	11	5	10

Figure 1. The magic square (4×4) [4]

1	6	5
8	4	0
3	2	7

Figure 2. Order 3 of magic square [6]

A Latin square design is an approach of mapping elements to appear in an equal form into a square matrix. Elements appear one time in each row and column. The processes are assigned at random into the square, with each process appearing one time per row and column [7]. No algorithm is used to build all kinds of magic square. Thus, the algorithm that executes for even squares is different from the algorithm that works on odd order.

There are three methods for constructing magic squares according to the matrix dimensions are those of the odd order, singly even order and doubly even order [8]. *Magic Squares of the Odd Order* a simple type of magic square is of the form 2m+1, where m is a positive integer. The De la Loubère's method is an example of an odd order, in which the matrix size may be 3×3, 5×5, and 7×7, amongst others.

Odd order magic squares were builded by using different methods, such as the pyramid, de la Loubere's, or staircase method [9]. *Magic Squares of a Doubly Even Order* the order of the doubly even ordered squares is of the form 4n (e.g., 4, 8, …) or may be divided by 2 and 4. An example is the method developed by Albrecht Dürer. The size of the square matrix is 4×4, 8×8 and 12×12, amongst others [10]. *Magic Squares of a Singly Even Order*. Singly even square in the order n is of the form 2(2m+1) =4n+2 (e.g., 2, 6, 10, 14, 18, 22, amongst others). The order can be divided by 2 but not 4. An example of this order is Philippe de la Hire's method. The size of the matrix is 6×6, 10×10 and 14×14, amongst others [11].

Different approaches used to construct magic squares have been developed during the past years. An example is the *dotting method*, which depends on cells marked by dots for the magic square. To construct a 4×4 magic square, dots are first placed on the main diagonals. Thereafter, the cells are computed from a corner and numbers are written in every marked cell. When the last cell is reached, the cells are reviewed in reverse and the numbers are placed sequentially in each cell without dots. Figure 3 illustrates the arranged dots in the square. The magic square (12×12) is obtaiend via computing the cells starting from a corner and ending in the opposite corner see Figure 4 [12]. All mathematical operations used in the proposed method were performed on a polynomial numbers of degrees GF (2^8).

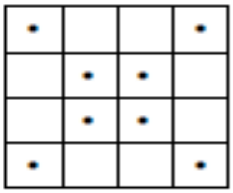


Figure 3. Arranged dots in the square

1	*143*	*142*	**4**	*5*	*139*	*138*	**8**	**9**	*135*	*134*	**12**
132	**14**	**15**	*129*	*128*	**18**	**19**	*125*	*124*	**22**	**23**	*121*
120	**26**	**27**	*117*	*116*	**30**	**31**	*113*	*112*	**34**	**35**	*109*
37	*107*	*106*	**40**	**41**	*103*	*102*	**44**	**45**	*99*	*98*	**48**
49	*95*	*94*	**52**	**53**	*91*	*90*	**56**	**57**	*87*	*86*	**60**
84	**62**	**63**	*81*	*80*	**66**	**67**	*77*	*76*	**70**	**71**	*73*
72	**74**	**75**	*69*	*68*	**78**	**79**	*65*	*64*	**82**	**83**	*61*
85	*59*	*58*	**88**	**89**	*55*	*54*	**92**	**93**	*51*	*50*	**96**
97	*47*	*46*	**100**	**101**	*43*	*42*	**104**	**105**	*39*	*38*	**108**
36	**110**	**111**	*33*	*32*	**114**	**115**	*29*	*28*	**118**	**119**	*25*
24	**122**	**123**	*21*	*20*	**126**	**127**	*17*	*16*	**130**	**131**	*13*
133	*11*	*10*	**136**	**137**	*7*	*6*	**140**	**141**	*3*	*2*	**144**

Figure 4. Constructed magic square (12×12) [12]

1.2. Finite fields

Finite fields are a collection of finite elements also called Galois fields (GF), which was named after Evariste Galois (1811-1832). Galois studied the scope of polynomials and discovered many of their principles. Numerous applications have used finite fields, such as cryptographic algorithms (Diffie and Hellman, 1976; ElGamal, 1985; Miller, 1986; Kravitz, 1993) and advanced encryption standard (AES) [13].

The finite fields executed for the order F_2^n is defined as GF (2^n) as a set of 2^n elements. The two binary operations + and × are defined in this set. Each nonzero element of the field has a multiplicative inverse [14]. All operations of finite fields produce an element in the field arrangement based on the p^n, p indicates a prime number and n indicates a positive integer [15]. One of the cases in finite fields is when the prime $(p) = 2$, in which the elements of GF (2^n) are expressed as binary numbers. One of the uses of GF (2^n) is a polynomial. $f(x)$ of polynomial number in GF (2^n) is showed in (1), which can be represented uniquely as n binary coefficients ($a_{n-1}a_{n-2} \ldots a_0$) [14].

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 = \sum_{i=0}^{n-1} a_i x^i \ldots \quad (1)$$

Finite field multiplication represents multiplying two polynomials elements and sums them like powers of x as a result. When multiplication result greater than n-1, so, the result is minimized via the module of irreducible polynomial m(x) of grade n then the result divided by $m(x)$ and kept the remainder [14]. "An irreducible polynomial is a polynomial $f(x)$ over a field GF (2^n), if and only if $f(x)$ can not be expressed as a two polynomials product, both over GF (2^n) and both of degree lower than of $f(x)$". Moreover, a polynomial is irreducible when it is divisible by itself and 1 (without remainder) [14, 16].

1.3. Related works

a) Dharini (2014) proposed the encryption methods RSA for secure data transmission, in which SSL over RSA and combined magic square provide additional security to the system; moreover, the confidentiality and integrity of data sent to and from the cloud are ensured [17].
b) Chenglian Liu (2011) proposed a novel approach of streams cipher application for random access file that can be easily implemented according to the magic square method, also improve the model of the cipher stream to strengthen the protection efficiently and has a high speed of key stream generator [18].

c) Shahla (2017) proposed an approach to creating a magic square of order 32, which represents the difficulty in tracing this square in the cryptography and improving efficiency by providing robust security to the encryption. The magic squares have numerous random numbers rather than the ASCII values and are used to generate the keys of the public-key encryption algorithms [19].
d) Authors (2013) this paper aim to enhance an algorithm to eliminate the repetitive characters or symbols in the ciphertext by using extra algorithms such as Function Encryption, NJJSAA, Bit Rotation, and Reverse method, the encryption is very difficult to break and more secure [3].
e) Ako (2016) the author has proposed a new method that combines encryption and information hiding to increase security, privacy, accuracy, and confidentiality. A hash least significant bit method has been suggested for the hiding encryption data process with the use of an affine cipher to provide more encryption and increase data security in the network environment [20].
f) Authors (2016). proposed approach is used that combines the encryption method with a method of hiding encrypted data to increase the security and maintain confidentiality, integrity, and availability of data against external attacks and unauthorized access. The plain text was encrypted with the RSA algorithm and the ciphertext was hidden into the image using the advanced LSB method. Where the plain text was encoded and divided into parts P1 and P2, the XOR operation was performed on the part (P1) of the odd locations and (p2) using even location for LSB+1 [21].
g) Jeena Pappachan, Jinu Baby (2015), suggested one of the kinds of chaotic maps (Tinkerbell Maps) with the magic square encrypt the images. The proposed method provides efficiency and security for encryption images.the proposed method consists of a 128-key secret key or a 16-character hexadecimal key that divides into 16 8-bit subkeys. The magic squares and two-dimensional maps are created, row shifting, pixel adjustment [22].
h) The authors (2012) proposed encryption grayscale and color images method using a symmetric-key Latin square image cipher (LSIC). this method improved novel Latin square image encryption and a novel method of merging probabilistic encryption in image encryption by including random noise by LSB technique. The proposed LSIC has a secure cipher due to large keyspace, excellent confusion, and diffusion approach and powerful against channel noise and brute-force attacks [23].
i) Al-Hasan (2018), proposed a new steganography approach is proposed by converting the cover image from RGB color space to YCbCr color space. Then, hide the encrypted data using the Affine Cipher and Magic Square Matrix are applied to embed the encrypted data onto the cover image using the ISB approach. Then the salt-and-pepper noise is added to the cover image. The results show that the proposed method withstands against attacks [24].
j) Tomba I (2017), an improved cryptosystem by uses 5 pseudo letters {Au, Ea, Ee, Oo, Ou} in the sequence of 26 English letters. The proposed pseudo letters are using magic squares or any type of matrices in encryption and decryption operation. Using pseudo letters will affect the ASCII characteristics thereby will provide an additional layer of security of the improved cryptosystem [25].

2. PROPOSED METHODOLOGY

The proposed encryption method contributes to security enhancement and eliminates the symbols repetition of ciphertext by using many new methods such as magic square, Latin square to encryption data and constructing random magic square to hiding ciphertext inside it, which provides additional safety features. The majority of the encryption methods suffer from the repetition of the elements into the ciphertext. So, random magic squares were used to hiding the repetition in the ciphertext. In the proposed method, the encryption and decryption processes were executed in two phases. The first phase is done by using a 3×3 Latin square derived from the odd ordered 3×3 magic square to arrange the 9 key elements of the polynomial numbers of degree GF (2^8). the plaintext of the 9-byte polynomial numbers is arranged in odd ordered 3×3 magic square and multiplying with Latin square 3×3 using the operations of finite fields with an irreducible polynomial of GF (2^8). The second phase involves constructs magic square of a doubly even order 12×12 includes 16 sub-squares of even magic square to hide the encrypted text of the first phase randomly. Figure 5 presents the block diagram of the encryption operation. The coding scheme alone does not provide sufficient security. Thus, magic square provides the permutation of the character encoding based on the magic square schemes.

2.1. Encoding

All plain text were labeled using the ASCII code for coding the letters to polynomial numbers of GF (2^8). Thereafter, the numbers were arranged in 3×3 matrix [PO] based on podision. Then the encoded numbers of plaintext are arrang in a 3 3 matrix [p] based on the construction of the odd order 3×3 magic square (M).

$$[PO] \rightarrow [P]$$

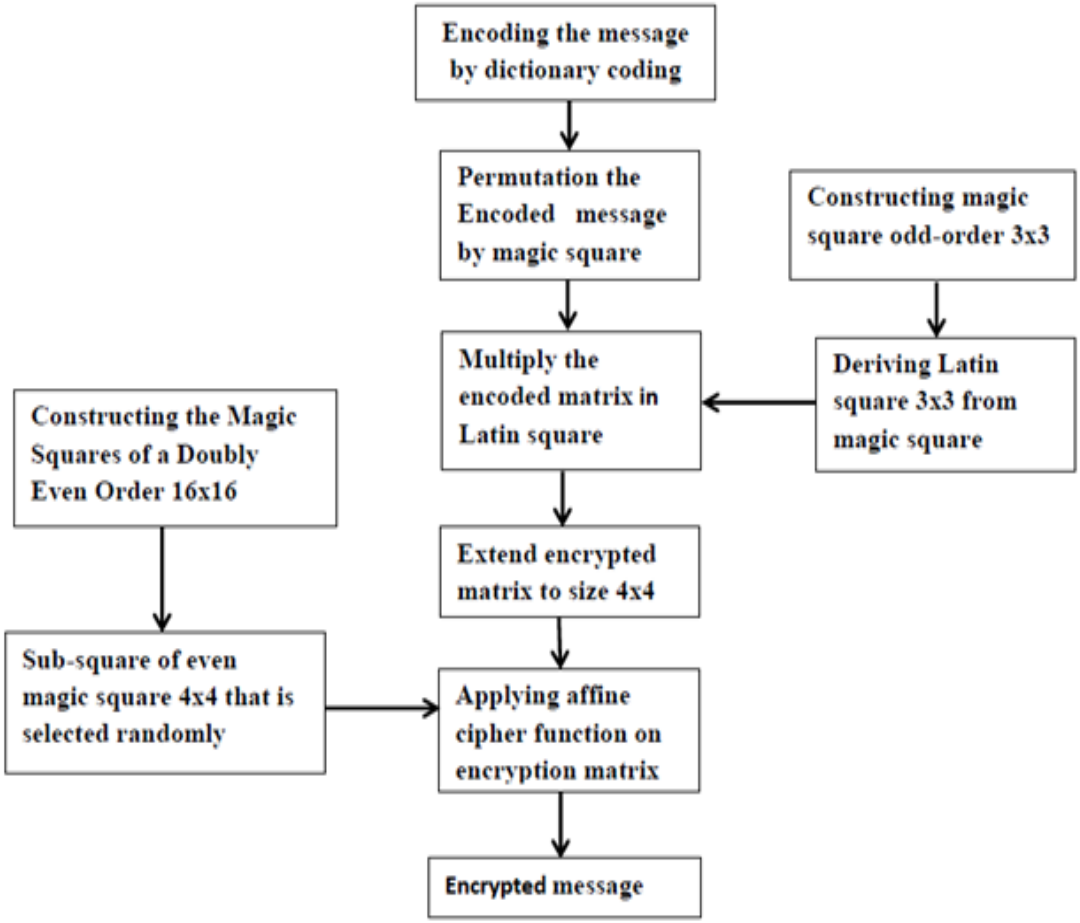


Figure 5. The block diagram of encryption operation

2.2. Encryption

The encryption process includes two phases of encryption and hiding the encrypted text.

Phase 1: The encoded elements of the plain text were arranged in a 3×3 matrix [p], and the key elements are arranged in an odd order 3×3 magic square [M]. Thereafter, the 3×3 Latin square [L] was derived from the magic square by taking modulo 3 as (2). The encrypted text was executed by multiplying the 3×3 Latin square [L] and encoded matrix [P] with an irreducible polynomial (m) ($x^8 + x^4 + x^3 + x + 1$) as (3).

$$L_i = M_i mod\ 3 \tag{2}$$

$$[C] = ([L].[P]) mod\ m \tag{3}$$

where

[C]: The encrypted matrix 3x3

Phase 2: The encrypted matrix [C] of the previous stage was randomly hidden inside the magic square [MS] of the 12×12 doubly even order. [MS] was divided into 16 by sub-magic square 4×4. The encrypted matrix [C] resulting from the previous stage is expanded to a 4 × 4 by randomly adding 8 numbers (salt) from 1 to 255 (polynomial numbers of GF (2^8)). Thereafter, the matrix [C] was imultiplied with one of the sub-magic squares [A] (selected randomly) by the affine cipher function. Each element of the encryption matrix C_i corresponds to one element in the sub-magic square A_i. The affine cipher as in (4) was applied, except one element that is left as a pointer. The result is a matrix [E] 4×4 represents the final of cipher text.

$$E_i = (A_i.C_i + b) Mod\ m \tag{4}$$

The Encryption algorithm of proposed method as the following:

Input: block (9 bytes) of encoded original data
Output: block (16 bytes) of cipher text

Step 1 Plain text [PO] was labeled using the ASCII code for coding the letters to polynomial numbers of GF (2^8) [P].

$$[PO] \rightarrow [P]$$

Step 2 The key elements were arranged in an odd order 3×3 magic square [M]. Latin square [L] was derived from the magic square by taking modulo 3. by multiplying the 3×3 Latin square [L] and encoded matrix [P] with an irreducible polynomial (m)

$$L_i = M_i mod\ 3 \quad (2)$$

$$[C] = ([L].[P]) mod\ m \quad (3)$$

[C] Expandes to a 4 × 4 by randomly adding 8 polynomial numbers of GF (2^8).

Step 3 Magic square [MS] of the 12 × 12 doubly even order was constracted. [MS] divides into 16 by sub-magic square4 × 4. The matrix[C] was multiplied with one of the sub-magic squares [A] (selected randomly) by the affine cipher function.

$$E_i = (A_i.C_i + b) Mod\ m \quad (4)$$

2.3. Decryption

The decryption process was executed by reversing the previous stages of the encryption process. Phase 1: In this phase, the encryption matrix [c] was returned by (5), where uses the inverse of each element of the specific sub-matrix [A] multiply with final encryption matrix elements [E]. The sub-magic square [A] was selected using the element (pointer) into the matrix. Then, the encryption matrix [C] was reduced from 4×4 to 3×3 by eliminating the random numbers (salt).

$$[C] = A_i^{-1}(E_i - b) mod\ m \quad (5)$$

Phase 2: This stage uses the inverse of 3×3 Latin squares [L^{-1}] and multiplied by the cipher matrix [C] to obtain the encoded plain text matrix [P]as in (6).

$$[P] = ([L^{-1}].[C]) mod\ m \quad (6)$$

Phase 3: Decoding the encoded plaintext matrix [P] to the plain text matrix 3×3 [PO] based on the positions of magic square elements [M].

$$[P] \rightarrow [PO]$$

The Decryption algorithm of proposed method as the following:

Input : block (16 bytes) of cipher text
Output : block (9 bytes) of plain text

Step 1 the inverse of each element of the specific sub-matrix [A] multiply with final encryption matrix elements [E].

$$[C] = A_i^{-1}(E_i - b) mod\ m \quad (5)$$

Step 2 the inverse of 3×3 Latin squares [L^{-1}] and multiplied by the cipher matrix [C] to obtain
the encoded plain text matrix [P].

$$[P] = ([L^{-1}].[C]) mod\ m \quad (6)$$

Step 3 Decoding the encoded plaintext matrix [P] to the plain text matrix 3×3 [PO] based on the positions of magic square elements [M].

$$[P] \rightarrow [PO]$$

Example

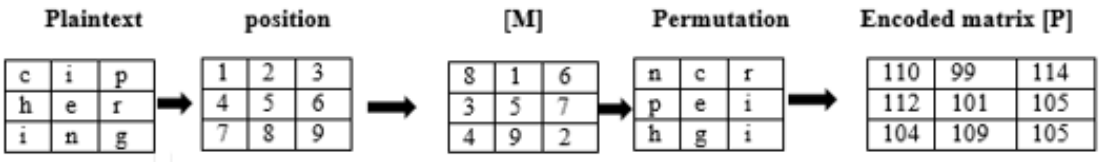
The message to be sent is assumed to be "Ciphering" and the key is a magic square “35, 37, 39, 41, 43, 45, 47, 49 and 51’, the steps of the encryption process as the follows:

a) The message of the plain text is encoded using the ASCII code as a Figure 6.

plaintext	c	i	p	h	e	r	i	n	g
cod	99	105	112	104	101	114	105	110	103
position	1	2	3	4	5	6	7	8	9

Figure 6. Encoding the message

b) One odd-order 3×3 magic square (M) is constructed for the permutation of the elements of the plain text based on the position of elements in (M), thereby obtaining the encoded matrix (P).



c) An odd order 3×3 magic square is constructed and a 3×3 Latin square (L) is derived from the MS base on mod 3, and multiplied in the encoded matrix with irreducible polynomial ($x^8+x^4+x^3+x+1$) as (7) and (8).

$$L_i = M_i \bmod 3 \quad (7)$$

[M]

49	35	45
39	43	47
41	51	37

Mod 3 =

[L]

1	2	0
0	1	2
2	0	1

$$[C] = ([L].[P]) mod\ m \quad (8)$$

[L]

1	2	0
0	1	2
2	0	1

*

[P]

110	99	114
112	101	107
104	109	105

=

[C]

142	169	160
160	191	187
180	171	141

d) The encrypted matrix is extended to 4x4 (Cx) by randomly adding 8 polynomial numbers of GF (2^8).

Cx =

243	142	169	160
231	160	191	187
150	180	171	141
12	7	**122**	**199**

e) Building The Magic Square doubly even order 16×16 including 16 sub-square of even magic square is as in Figure 7.

1	255	254	4	5	251	250	8	9	247	246	12	13	243	242	16
240	18	19	237	236	22	23	233	232	26	27	229	228	30	31	225
224	34	35	221	220	38	39	217	216	42	43	213	212	46	47	209
49	207	206	52	53	203	202	56	57	199	198	60	61	195	194	64
65	191	190	68	69	187	186	72	73	183	182	76	77	179	178	80
176	82	83	173	172	86	87	169	168	90	91	165	164	94	95	161
160	98	99	157	156	102	103	153	152	106	107	149	148	110	111	145
113	143	142	116	117	139	138	120	121	135	134	124	125	131	130	128
129	127	126	132	133	123	122	136	137	119	118	140	141	115	114	144
112	146	147	109	108	150	151	105	104	154	155	101	100	158	159	97
96	162	163	93	92	166	167	89	88	170	171	85	84	174	175	81
177	79	78	180	181	75	74	184	185	71	70	188	189	67	66	192
193	63	62	196	197	59	58	200	201	55	54	204	205	51	50	208
48	210	211	45	44	214	215	41	40	218	219	37	36	222	223	33
32	226	227	29	28	230	231	25	24	234	235	21	20	238	239	17
241	15	14	244	245	11	10	248	249	7	6	252	253	3	2	256

Figure 7. Magic square doubly even order 16×16

f) By applying the affine cipher function on encryption matrix (C) and sub-square of the even 4×4 magic square [A] selected randomly, the result represents the final encrypted matrix. The affine cipher function on each element of the encryption matrix is applied, except the element at the bottom of the left corner is substituted with an element at the bottom left corner in the sub-magic square to be a pointer, where (b) value is equal 180 as a constant in (9).

$$E_i = (A_i . Cx_i + b) Mod\ m \quad (9)$$

A=

1	255	254	4
240	18	19	237
224	34	35	221
49	207	206	52

Cx =

243	142	169	160
231	160	191	187
150	180	171	141
12	7	**122**	**199**

E =

71	**92**	**236**	**2**
141	**1**	**107**	**225**
60	**182**	**238**	**87**
227	**239**	**174**	**52**

2.4. The decryption process uses two stages as follows:

g) The inverse of the affine cipher function on the encryption matrix (E) and the specific sub-inverse magic square affine cipher is applied as following in inverse (10):

$$Cx = A^{-1}(E - b) mod\ m \quad (10)$$

A^{-1} =

1	28	65	203
91	170	75	80
177	90	241	248
69	230	172	52

E =

71	**92**	**236**	**2**
141	**1**	**107**	**225**
60	**182**	**238**	**87**
227	**239**	**174**	**52**

Cx =

243	142	169	160
231	160	191	187
150	180	171	141
12	7	**122**	**199**

h) The matrix is returned to 3×3 [c] and the random numbers (salt) that added at the encryption stage were deleted.

$Cx =$

243	142	169	160
231	160	191	187
150	180	171	141
12	7	**122**	**199**

⟹

$C =$

142	169	164
160	191	185
180	173	141

i) The inverse of the Latin square is used and multiplied with the 3×3 encryption matrix as following in (11).

$$P = ([L^{-1}].[C]) Mod\ m \quad (11)$$

$L^{-1} =$

79	**158**	**39**
39	**79**	**158**
158	**39**	**79**

$C =$

142	169	164
160	191	185
180	173	141

$P =$

110	99	114
112	101	107
104	109	105

3. RESULTS AND ANALYSIS

The proposed method provided the processing of a repetition problem in the ciphertext. The important characteristic of the encryption algorithm is the Avalanche Effect, where any single bit change in the plain text or key must be a more change in the bits of encrypted text. Avalanche effect test of the proposed method can be calculated by using (12). The test calculates the avalanche effect of the proposed method when changing 1-bit in the key and keeping plaintext constant and changing in plaintext by 1-bit and keeping the Key bits constant. In two cases, the number of bits that differ between the two cipher-texts is calculated by XOR operation. Then, Table 1 shows the avalanche effect of proposed method:

$$Avalanche\ Effect = \frac{\text{Number of chang bits in the ciphered text}}{\text{Number of bits in the ciphered text}} \times 100\% \quad (12)$$

Table 1. Shows the avalanche effect of proposed metgod

Sample No.	Keys	input	Output MDES	Avalanche test	Avalanche Test %
1	49, 35, 45, 39, 43, 47, 41, 51, 37	99, 105, 112, 104, 101, 114, 105, 110, 103	71, 92, 236, 2, 141, 1, 107, 225, 60, 182, 238, 87, 227, 239, 174, 52	0.648	64.8 %
	49, 35, 45, 39, **171**, 47, 41, 51, 37	99, 105, 112, 104, 101, 114, 105,110, 103	255, 242, 24, 111, 282, 255, 158, 62, 210, 32, 161, 180, 250, 49, 251, 171		
2	18, 155, 80, 228, 110, 160, 14, 51, 169	67, 111, 109, 112, 117, 116, 101, 114, 115	108, 157, 124, 219, 140, 170, 151, 191, 20, 113, 224, 181,184, 253, 200, 70	0.609	60.9%
	18, 155, 80, 228, 110, 160, 14, 51, 169	67, 111, 109, **240**, 117, 116, 101, 114, 115	156, 105, 243, 139, 233, 217, 254, 67, 250, 191, 26, 90, 150, 1, 50, 212		
Main percentage avalanche value				0.628	62%

$$Avalanche\ Effect\ Sample\ 1 = \frac{4+5+5+5+4+7+6+7+6+4+4+5+5+6+4+6}{128} = \frac{83}{128} = 0.648$$

$$Avalanche\ Effect\ Sample\ 2 = \frac{4+5+5+2+4+5+4+6+6+5+6+7+4+6+6+3}{128} = \frac{78}{128} = 0.609$$

The Avalanche Effect results of the proposed method indicate that the ratio of 62% is a good ratio, where the accepted ratio is 50% indicates the algorithm has perfect confusion and diffusion, as well as the ratio indicates that there is less repetition in the ciphertext When the blocks of plain text are almost the same. This good ratio comes from that method when it is selected the sub-magic square randomly in each encryption process. The randomness tests are important for testing the cipher text to determine if there is any deviation or biases between plaintext/cipher text bits and to ensure the random form for the cipher text. The proposed technique provides accepted results based on the NIST randomness tests. Where a P-value for a test is equal to one, and then the value will be ideal randomness. A P-value of zero refers that the value is

completely non-random. The results tested are displayed in Table 2. The results of the randomness tests of the proposed method are acceptable for all the p-values of the statistical tests where all p-values are nearest from 1 that indicated the encrypted data are random text. Accordingly, that randomness results are acceptable and no frequency of all possible overlapping m-bit patterns across the entire sequence.

Table 2. Proposed method results of the NIST randomness tests

	Statistical Tests	Input Size (n)	P-value Proposal method	The results
1	Frequency (monobit)	10000	0.873124	Pass
	Test	100000	0.624278	Pass
	Average of P-value		**0.748701**	Pass
2	Block Frequency	10000	0.287645	Pass
	(m=8)	100000	0.634265	Pass
	Average of P-value		**0.460955**	Pass
3	Approximate Entropy	10000	0.876541	Pass
	Test m=3	100000	0.955915	Pass
	Average of P-value		**0.916228**	Pass
4	Linear Complexity	10000	0.447321	Pass
	Test (M=100)	100000	0.505732	Pass
	Average of P-value		**0.476526**	Pass
5	Runs Test	10000	0.289120	Pass
		100000	0.801208	Pass
	Average of P-value		**0.545164**	Pass
	Total Averages of P-value		0.629515	Pass

4. CONCLUSION

The proposed technique process the problem of repetition in the ciphertext and increase the permutation of the plain text by uses the functionality of a 3×3 magic square and the diagonal Latin square. The proposed method demonstrated the ability to process repetition in encrypted data effectively when entering similar texts in each encryption process, in addition, the method demonstrated the high randomness of the encrypted texts. The proposed was achieved by uses a magic square 12×12 to hiding encrypted data and eliminates the repetition elements. Thus, the encrypted data are substantially secure and robust against attackers based on the tests the Avalanche Effect and randomness results that shown accepted results in the ciphertext. It is recommended that research is used in the field that corresponding database encryption because the database security system needs encryption methods that not suffer from the problem of repetition data.

REFERENCES

[1] Das, Debasis, U. A. Lanjewar, and S. J. Sharma, "The Art of Cryptology: From Ancient Number System to Strange Number System," *International Journal of Application or Innovation in Engineering& Management (IJAIEM)*, vol. 2, no. 4, pp. 265-275, 2013.

[2] Stallings, William, "Network Security Essentials: Applications and Standards Fourth Edition," *Prentice Hall, 1 Lake Street, Upper Saddle River, NJ*, New Jersey, 2011.

[3] Danti, Ajit, and Rajesh Nayak, "Data Encryption by Excluding Repetitive Character in Cipher Text," *International Journal Of Innovationsin Engineering And Technology (IJIET)*, vol. 2, no. 4, pp. 270-276, 2013.

[4] Ess, Eli, "The Magic of Magic Squares," Thesis in Mathematics Middlebury College, 2005.

[5] N. Shibiraj, Tomba," Modified Hill Cipher: Secure Technique using Latin Square and Magic Square," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 12, pp. 315-320, 2018.

[6] P. Bartlett, "Latin Squares and Magic," Mathcamp, 2012.

[7] Gao, Lei, "Latin squares in experimental design," Michigan State University, 2005.

[8] D. I. George, J. Sai Geetha and K. Mani, "Add-on Security Level for Public Key Cryptosystem using Magic Rectangle with Column/Row Shifting," *International Journal of Computer Applications*, vol. 96, no. 14, pp. 38-43, 2014.

[9] Nitin Pandey, D. B. Ojha, "Secure Communication Scheme with Magic Square," *Journal of Global Research in Computer Science*, vol. 3, no. 12, pp. 12-14, 2012.

[10] P. K. Srinivasan, "Mathematics and Magic Square," *Magic Square*, 1992.

[11] Ronald P. Nordgren, "New Constructions For Special Magic Squares," *International Journal of Pure and Applied Mathematics*, vol. 78, no. 2m pp. 133-154, 2012.

[12] Sesiano, Jacques, "Magic Squares: Their History and Construction from Ancient Times to AD 1600," *Springer International Publishing*, 2019.

[13] Ali, Nada Hussein M., and Suaad Ali Abead, "Modified Blowfish Algorithm for Image Encryption using Multi Keys based on five S-boxes," *Iraqi Journal of Science*, vol. 57, no. 4C, pp. 2968-2978, 2016.

[14] Stallings, William, "Cryptography and network security: principles and practice 6 Edition," *Person Education Inc*, 2014.

[15] Madhuri, O. B. B., E. Rambabu, and M. Malijeddi, "Design and Implementation of Arithmetic Unit for GF (2m)," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 1, no. 9, 2012.
[16] Welschenbach, Michael, "Cryptography in C and C++," *Apress*, 2006.
[17] Dharini, A., R. M. Saranya Devi, and I. Chandrasekar, "Data Security for Cloud Computing Using RSA with Magic Square Algorithm," *International Journal of Innovation and Scientific Research*, vol. 11, no. 2, pp. 439-444, 2014.
[18] Liu, Chenglian, et al., "A Study on the Stream Cipher Embedded Magic Square of Random Access Files," *AIP Conference Proceedings*, vol. 1389, no. 1, pp. 1070-1073, 2011.
[19] Umar, Shahla Uthman, "An Improved RSA based on Double Even Magic Square of order 32," *Kirkuk University Journal /Scientific Studies (KUJSS*), vol 12, no. 4, pp. 1-18, 2017.
[20] Abdullah, Ako Muhammad, and Roza Hikmat Hama Aziz, "New approaches to encrypt and decrypt data in image using cryptography and steganography algorithm," *International Journal of Computer Applications*, vol. 143, no. 4, pp. 11-17, 2016.
[21] Alhassan, John K., et al., "A Secure Method to Hide Confidential Data Using Cryptography and Steganography," *International Conference on Information and Communication Technology and Its Applications (ICTA 2016)*, pp. 105-110, 2016.
[22] Jeena Pappachan, Jinu 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.
[23] Wu, Yue, et al., "A novel latin square image cipher," *arXiv preprint arXiv: 1204.2310*, 2012.
[24] Al-Hasan, Waleed S. Hasan, Muhammad Mun'im Ahmad Zabidi, and Ab Al-Hadi Ab Rahman, "A New Steganography Technique Using Magic Square Matrix and Affine Cipher," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 2-8, pp. 119-125, 2018.
[25] Tomba, I., "Add-on security using Weak Magic squares in Public Key Cryptosystem, Modified with Dummy Letters*," International Journal of Scientific and Research Publications*, vol. 7, no. 2, pp. 257-261, 2017.

BIOGRAPHIES OF AUTHORS

Sahab Dheyaa was born in Baghdad in 1960. He has a Ph.D. in computer science-security from Informatics Institute for Postgraduate Studies Baghdad, Iraq.has M.Sc. in computer science from the College of Science AL-Mustansiriyah University from 2003- 2005. From 1997-2000 he received his B.Sc. in Computer Science-Computer Dept. College of Education, AL-Mustansiriyah University. From 1980 to 1984, he received also his B.Sc in administration science in economic and administration College, AL-Mustansiriyah University. He is a lauctuelar in University of Information Technology and Communications. Current research interests: network security, database security, Image processing

.

Taha Mohammed Hasan received his BSc, MSc in computer science from Mansour University College and The University of Mustansiriyah, Baghdad, Iraq in1992 and 2006 respectively and PhD in computer science From Harbin Institute of Technology (HIT), Harbin, China in 2013. Currently he is an assistant professor in University of Diyala College of Sciences Computer Science Department, Iraq. His research interests include image processing, pattern recognition, cloud computing and biometrics, etc