A New Block Cipher Algorithm Using Magic Square of Order Five and Galois Field Arithmetic with Dynamic Size Block

https://doi.org/10.3991/ijim.v15i16.24187

Ibrahim Malik ALattar(✉), Abdul Monem S. Rahma
University of Technology, Baghdad, Iraq
ibrahiminter@yahoo.com

Abstract—This paper describes the development of encryption algorithms using the magic square of order 5 and Multi-level keys with the addition of Matrix keys to increase implementation speed and complexity. This work relied mainly on the magic sum and some equations that were added as an improvement on previous work. Multi-level keys were used for three different message sizes, and an additional key matrix with size 5×5 was used to add more complexity. The proposed work was performed using both GF(P) and GF(2^8). Results were compared with the MS3, they have been found good, with acceptable speed and high complexity where it was $(P)^9 \times (256)^{16}$ in the first algorithm, $(P)^9 \times (256)^{16} \times 3$ in the second algorithm, and $(P)^9 \times (256)^{16} \times 3 \times (P)^{25}$ in the third algorithm, the complexity changed according to the chosen value of N randomness, in addition to speed, complexity, NIST calculations have been performed for texts and histogram calculations for different images were calculated and compared as well.

Keywords—cryptography, GF(2^8), GF(P), magic square, multi-level key

1 Introduction

The need for modern society to maintain data security has led researchers, mathematicians, and cryptologists to develop encryption algorithms and make them more complex and difficult to access to ensure the protection of information [1]. Interest in magic squares dates back to ancient times, where it was found that BC. Jugglers and magicians used them in their incantations; the Islamists in the past were very interested in magic squares too, as it was previously believed that they contained the nine letters that were revealed to Prophet Adam [2]. Mathematicians and cryptography scholars are also among the first ones interested in magic squares, where many intelligence games have been built based on the idea of magic squares, such as Sudoku [2]. Where the magic square of the third degree was used by assigning some positions to the key and others to the message and multiplying the resulting matrix (from the combination of the key and the message) with a second key matrix equal to the size of the magic square used. As it relied mainly on magic constant [3]. The work was previously developed using the magic square of fifth-degree and different key lengths have been used, namely; 10 and

14 and the key lengths were 15 and 11 respectively. The work has been mainly based on the magic sum [4]. the most important previous works are reviewed through which magic squares were used. In the year 2015, Dawood et. al proposed a new algorithm based on the mathematical foundations of the magic square and the magic cube, where the Diffie-Hellman was used to determine the dimensions of the magic square [5]. Also, a group of researchers developed an algorithm using the magic square, in which two specific magic squares were used and created, then they have used it in cryptography [6]. In 2016, Dawood et al. developed a new method by building the magic cube by folding the magic square, and this method was generalized regardless of the type of magic square used [7]. In 2017, Umar proposed a method based mainly on the 32nd magic square, whereby the duplication of characters was eliminated and the encryption efficiency was improved [8]. In 2019, Al-Hashemy et al. proposed a new algorithm for encrypting images using the magic square, where a random key generation was used and the XOR operation was employed [9]. In the same year, Jabbar et al. proposed a protocol using Magic Square of order 3 and used it in encryption [3]. In 2020, Kareem and Rahma introduced an amendment to the Twofish algorithm in cryptography, whereby 2 keys were used with several rounds, and multi-level keys were used for 4 different message sizes during the work in addition to the use of GF [10]. Later in 2021, Cuevas and Netzer proposed and developed a method based on magic squares and their relationship to cryptography and quantitative information and their relationship to the games [11].

2 Previously technologies and advantages

Magic squares have many features, so the squares in which the sum of each column is equal to the sum of each row in it is called mini magic square, while if the sum of each row, column, and equal diameter is called normal magic square while if the numbers in the magic square are all prime the square is then called Prime Magic Square[12,16].

To take advantage of the properties of the magic squares, the switch has been made from unspecified numbers to specific numbers, depending on Prime Number(P), and all Galois field (GF) will depend on P[12]. The need for the devices currently in use prompted the change from relying on GF(P) to the use of polynomial numbers GF(2^8), as each Galois field will depend on the irreducible polynomial in its mathematical that used addition and multiplication[13]. Just as for each specific GF(2^{Number}) there is a certain number of irreducible polynomials that can be used within that range, there are only 30 irreducible polynomials that can be used within GF(2^8) [14,17]. Before retrieving the original text, the work will be changed to a linear equation system and noting that the number of equations is equal to the number of unknowns to be solved. before solving the work by Gaussian elimination or anyway is chosen requires arranging the equations in such a way that the main diagonal does not contain the value zero, and the final result of the Gaussian elimination method will restore the plain text [15,18]. Cryptography is a science that uses mainly mathematics to encode and decode data by using algorithms with the help of a key [14].

Multi-level keys is a system that consists of more than one layer of keys, in such a way that the information is stored in a small database and the required key sequence is chosen from it [15]. In this proposed paper, we will use magic square, GF with both

types, Additional message length, Multi-level keys, and Extra Key, in order, as will be seen later.

3 The proposed cryptography technique

Since increasing the number of equations in the magic square leads to an increase in the speed of the encryption, a method for increasing the speed is therefore proposed.

3.1 Add more equations to the system proposed

Four additional equations were added to the proposed system as shown in Figure 1 below.

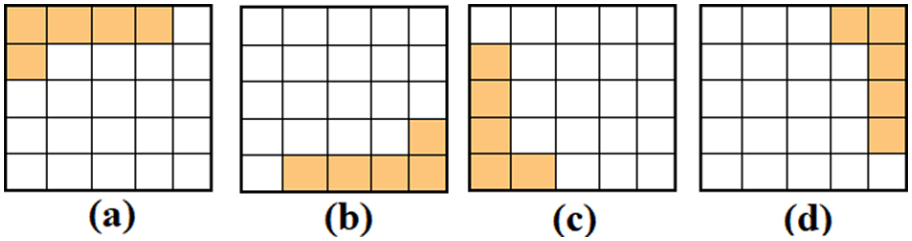

Fig. 1. The extra equations added to the work

By checking the above equations note that the latest two (Figure 1 c & d) are cause dependency with the overall equations, for that reason, they are canceled and remain added only two (Figure 1 a & b), so the total of equations will be equal to 16. Therefore, therewill be 16 locations for the message corresponding to 16 equations (out of 25 for MS 5) and the remaining 9 locations for a key. Which would like to refer to that there are no keys locations or key values determined by the specified algorithm, but rather they are flexible to the user.

3.1.1 Inserting the message and key. By default the following locations for the key were considered to be selected:

	k		k	
k				k
		k		
k			k	
	k			k

Fig. 2. Key locations to be selected for message length = 16

From Figure 2, where the shown positions represent the selected key locations (9 locations) and the remaining sites will represent the message positions in about 16 locations, so the total is 25 locations (for MS 5), as shown in Figure 3 below.

A 00	A 01	A 02	A 03	A 04
A 10	A 11	A 12	A 13	A 14
A 20	A 21	A 22	A 23	A 24
A 30	A 31	A 32	A 33	A 34
A 40	A 41	A 42	A 43	A 44

Fig. 3. The key (colored locations) while the remaining are the message locations

Therefore, sums are obtained as shown by the equations below:

$$
\begin{aligned}
e1 &: Sum1 = A_{00} + A_{01} + A_{02} + A_{03} + A_{04} \\
e2 &: Sum2 = A_{10} + A_{11} + A_{12} + A_{13} + A_{14} \\
e3 &: Sum3 = A_{30} + A_{31} + A_{32} + A_{33} + A_{34} \\
e4 &: Sum4 = A_{40} + A_{41} + A_{42} + A_{43} + A_{44} \\
e5 &: Sum5 = A_{00} + A_{10} + A_{20} + A_{30} + A_{40} \\
e6 &: Sum6 = A_{01} + A_{11} + A_{21} + A_{31} + A_{41} \\
e7 &: Sum7 = A_{03} + A_{13} + A_{23} + A_{33} + A_{43} \\
e8 &: Sum8 = A_{04} + A_{14} + A_{24} + A_{34} + A_{44} \\
e9 &: Sum9 = A_{00} + A_{11} + A_{22} + A_{33} + A_{44} \\
e10 &: Sum10 = A_{04} + A_{13} + A_{22} + A_{31} + A_{40} \\
e11 &: Sum11 = A_{02} + A_{11} + A_{20} + A_{34} + A_{43} \\
e12 &: Sum12 = A_{01} + A_{10} + A_{24} + A_{33} + A_{42} \\
e13 &: Sum13 = A_{02} + A_{13} + A_{24} + A_{30} + A_{41} \\
e14 &: Sum14 = A_{03} + A_{14} + A_{20} + A_{31} + A_{42} \\
e15 &: Sum15 = A_{10} + A_{00} + A_{01} + A_{02} + A_{03} \\
e16 &: Sum16 = A_{34} + A_{44} + A_{43} + A_{42} + A_{41}
\end{aligned}
\tag{1}
$$

The recipient will solve the 16 equations, where there will be 16 unknown corresponding to the 16 equations, and the remaining 9 are known to both sides (the key).

Before solving equations, it is required to arrange the equations in such a way that the principal diameter is not equal to zero. The equations will be solved according to the rules of the specified field to obtain the final solution of the text or image message (see sec. 3 Para. 4) in this paper. The proposed work was developed to increase the strength and difficulty of breaking it so that the polynomial number was used instead of integers in both the message and the key value.

3.1.2 The suggested algorithms. There is neither a fixed nor a specific algorithm, and the following suggested method represents an improvement for GF 5 both encryption and decryption respectively.

Algorithm 1-a: The Suggested Symmetric cipher algorithm For MS 5 (Encryption)

Input: Original message Or Image, key position, and key values.
Output: Ciphertext.
Begin:
Step1: The value of the keys is placed at the agreed positions in MS 5.
Step2: The remaining places will accommodate the message.
Step3: The result of Steps 1 and 2 will be that MS5 will be filled with 16 locations for the message and 9 positions for the key, and this result will represent the matrix A (as seen in Figure 3), and in this step will find the results for all summation in (1).
The final output for this step will be representing the ciphertext.
End.

Algorithm 1-b: The Suggested Symmetric cipher algorithm For MS 5 (Decryption)

Input: Ciphertex, key positions, and key values.
Output: Original text or Image.
Begin:
Step1: The agreed key value is placed at the agreed positions in MS 5.
Step2: The remaining positions will contain the message, which is equal to 16 positions that equal to the number of equations (number of sums), Depending on matrix A (Figure 3) and (1).
Step3: The equations obtained from Step 2 will be arranged so that the main diagonal does not contain the zero value.
Step4: The sixteen equations will be solved based on the finite domain rules used for GF(P), as seen in this paper (sec. 3 Para. 4).
Through this step, the original text or image will be restored.
End.

A development to the algorithm 1 suggested above was done to suit the devices currently in use and to increase the strength and the difficulty of breaking it, so the GF(P) was replaced by GF(2^8) because of the devices at present use 8 bits in their work.

3.2 Use message lengths together

Using the different longest messages together will increase the power of the proposed algorithm. In this proposed algorithm #2 will use Multi-level keys.

Therefore, the numbers 0, 1, and 2 are randomly chosen. If the selected number was equal zero, then it will be encoded and decoded using MS 5 with the message length = 10. Whereas if the selected number = one, the encryption algorithm of MS 5 will be used with message length = 14. Otherwise, encryption & decryption will be done by MS 5 with message length = 16. (Algorithm 1 in this paper). The Suggested algorithm 2 will be illustrated in Figure 4 below:

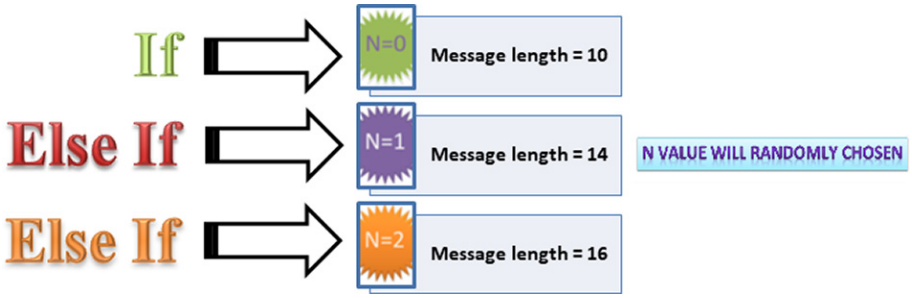

Fig. 4. Algorithm # 2 Suggested Multi-level keys for MS 5

3.3 Use an extra key

The present proposed algorithm represents Algorithm # 3 in this paper.

Thus, increasing the number of keys used will increase the strength of the encryption.

3.3.1 The process of multiply the key in MS 5. A square will be used as a 2D array whose size is equal to the size of MS 5, and the values inside will be randomly selected to act as a key between the two ends, as shown in Figure 5 below.

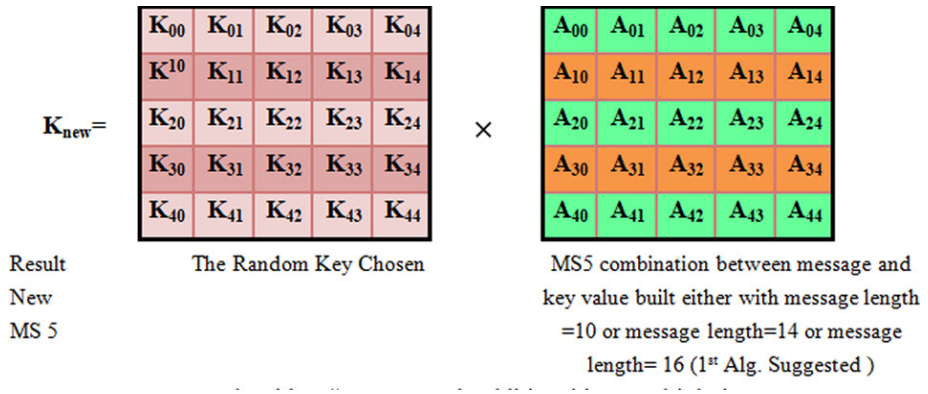

Fig. 5. Algorithm #3 Suggested Additional key Multiply by MS 5

3.3.2 The Suggest algorithm #3.

Algorithm 3-a: The Suggested Symmetric cipher algorithm For MS 5 (Encryption)

Input: Original message Or Image, keys position, keys values, keys Matrix, and Multi-level keys.
Output: Ciphertext.
Begin:
Step1: Randomly Choose n value to choose the length of the message (see Figure 4).
Step2: same as Step1+ Step2 in Alg.1 Encryption in this paper.
Step3: the result of Step2 will multiplyw with the key Matrix cell by cell (see Figure 5).
Step4: the result of the step3 will be treated like Step3 in Alg.1 Encryption in this paper. The result of step4 (the sums) will represent the ciphertext.
End.

Algorithm 3-b: The Suggested Symmetric cipher algorithm For MS 5 (Decryption)

Input: Ciphertext, keys position, keys values, keys Matrix, Multi-level keys used.
Output: Original text or Image.
Begin:
Step1: Depending on the value of the key N to find out the length of the message used (see Figure 4).
Step2: Same as Step1 in Alg.1 Decryption in this paper.
Step3: The remaining positions in MS 5 will filled with the message and here the length of the message will depend on the value of n.
Step4: Same as Step3+Step4 in Alg.1 Decryption in this paper.
Step5: The result from Step4 will find the inverse for each number with the key Matrix to find the original text or Image.
End.

In this algorithm 3#, development was also made, where the prime number was replaced by an irreducible polynomial to increase efficiency, power, reduce speed, and make the algorithm more compatible with devices.

4 Evaluation

Cryptography is a method used to maintain the confidentiality, integrity, and Authentication of data when it is sent between two parties that have a connection, the method is done by combining the sent data with a key (the key is known only between the two recipient parties) and sending it to the receiving party, on the opposite side, the recipient extracts the key from the encrypted text received to obtain the original text, this is done using mathematical methods. On the other hand, the number of attempts that the third party is supposed to try to access the sent data is called brute force attack. In this paper, MS5 is used with message length = 16, the use of multi-level keys for three lengths of the message, and the process of multiplying the key with MS5. This is all as an improvement on MS3 and MS4 by continuously increasing the complexity of the proposed algorithms in an ascending sequence to reach to state of difficult-to-access data. In this part, will discuss and compare (complexity, speed, histogram calculation of images, and NIST calculations) of the three proposed algorithms for GF with its two types, and compare the results for the proposed algorithms together, and also compare the results with MS3. The results in this section were obtained using a laptop with the following specifications:

Processor : Intel(R) Core(TM) i5-4310M CPU @ 2.70GHz (4 CPUs), ~2.7GHz.
Installed Memory (RAM) : 8 GB, System Type : 64-bit Operating System.

4.1 Security complexity for brute force attack

The complexity of the brute force attack is represented by the complexity of the key only for each proposed algorithm and for MS3. For MS3 using GF(P), the calculation of brute force attack power = 3, represented by the length of the key used and the prime number used = P while using GF(2^8) the key length is also 3 and $2^8 = 256$. While for the first algorithm proposed using GF(P), it is calculated by the remaining number of subtracting the message length for MS5 = 9 (25-16), and the prime number was used will be referred to by P as shown in (2), as if used irreducible polynomial in GF(2^8), also it is calculated with 9 (the remaining number From subtracting the length of the message) and $2^8 = 256$, see (3).

$$B3 = (p)^9 \tag{2}$$

$$B4 = (256)^9 \tag{3}$$

As for the second algorithm proposed, the difference from the first algorithm will be the multi-level keys used, which equals 3 multiplied by the results of the first algorithm (Depends on the length of the message(, for each of the two types of GF, see (4) and (5).

$$B5 = (p)^{length\ of\ message} \times 3 \tag{4}$$

$$B6 = (256)^{length\ of\ message} \times 3 \tag{5}$$

As for the third proposed algorithm for GF(P), the result of the second algorithm will be multiplied by the key matrix used, which is equal to the size of MS5 and the prime number = P, see (6), and for GF(2^8) will also the result for algorithm #2 proposed multiplied by the matrix Key and multiplied by the number of irreducible polynomials allowed used for GF(2^8), as shown in (7).

$$B7 = (P)^{length\ of\ message} \times 3 \times (P)^{25} \tag{6}$$

$$B8 = (256)^{length\ of\ message} \times 3 \times (256)^{25} \times 30 \tag{7}$$

And Figure 6 will show the results of a comparison of the results of the brute force attack of the three proposed algorithms with MS3.

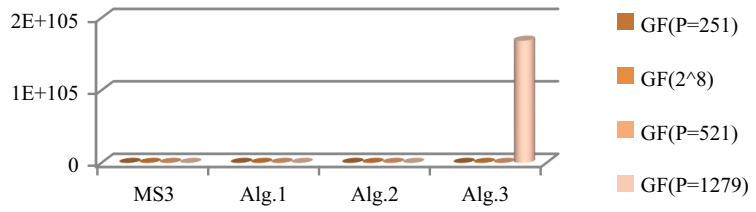

Fig. 6. Results for brute force attack for all algorithms using GF(P) and GF(2^8)

4.2 Security complexity analysis

Here will calculate the data complexity for each algorithm separately, For MS3, data complexity would be equal to ASCII code which is equal to 256 raised to the power of the message length would be 6 (size of general MS3 minus key length = 3. As for the first algorithm suggested, the data complexity will be the ASCII code number raised the message length used, which equals 16. As for the data complexity of the second algorithm suggested, it will have three states according to the message length used 10, 14, or 16, and the ASCII code itself is equal to 256, so the complexity in the data will be either $(256)^{10}$, $(256)^{14}$ or $(256)^{16}$ (depending on the length of the message). As for the data complexity of the third algorithm suggested, it will be quite similar to the second algorithm. Thus, overall complexity = keys complexity (brute force attack complexity multiplied by data complexity), each algorithm will be calculated separately in equations (8)–(21), and the results are presented and compared with MS3 as in Figure 7.

$$C1 = (p)^{9} \times (256)^{16} \tag{8}$$

$$C2 = (256)^{9} \times (256)^{16} \tag{9}$$

If N=0
$$C3 = (P)^{15} \times 3 \times (256)^{10} \tag{10}$$

If N=0
$$C4 = (256)^{15} \times 3 \times (256)^{10} \tag{11}$$

If N=1
$$C5 = (P)^{11} \times 3 \times (256)^{14} \tag{12}$$

If N=1
$$C6 = (256)^{11} \times 3 \times (256)^{14} \tag{13}$$

If N=2
$$C7 = (P)^{9} \times 3 \times (256)^{16} \tag{14}$$

If N=2
$$C8 = (256)^{9} \times 3 \times (256)^{16} \tag{15}$$

If N=0
$$C9 = (P)^{9} \times 3 \times (P)^{25} \times (256)^{10} \tag{16}$$

If N=0
$$C10 = (256)^{9} \times 3 \times (256)^{25} \times 30 \times (256)^{10} \tag{17}$$

If N=1
$$C11 = (P)^{11} \times 3 \times (256)^{11} \times (256)^{14} \tag{18}$$

If N=1
$$C12 = (256)^{11} \times 3 \times (256)^{11} \times (256)^{14} \tag{19}$$

If N=2
$$C13 = (P)^{9} \times 3 \times (P)^{25} \times (256)^{16} \tag{20}$$

If N=2
$$C14 = (256)^{9} \times 3 \times (256)^{25} \times 30 \times (256)^{16} \tag{21}$$

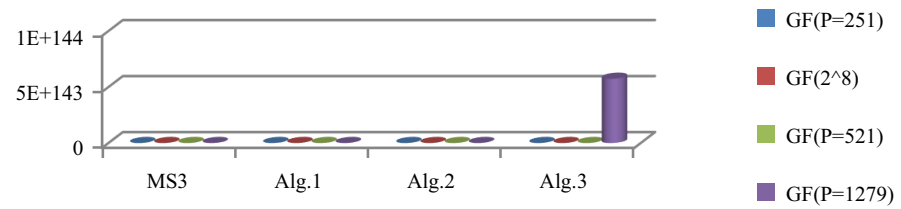

Fig. 7. The complexity for Suggested algorithms and MS3 with GF(P) and GF(2^8)

4.3 Time for encryption and decryption processes

Another measure that was calculated for the proposed algorithms is the time of implementation of the encryption and decryption where experiments were conducted on the text of different characters' length and images of different sizes, times were found for GF of both types, and Tables 1 & 2 clarify the measured times, and statistics were made and Comparison between them was done as shown in the Figures 8 & 9.

Table 1. The average time to implementation the suggested algorithms using texts

Algorithm Proposed	GF Type	Encryption Time	Decryption Time
Alg. #1	GF(P)	.00263200	.1035217
Alg. #1	GF(2^8)	.15375520	.4353350
Alg. #2	GF(P)	.00436370	.1000310
Alg. #2	GF(2^8)	.14820300	.2311280
Alg. #3	GF(P)	.00678400	.1048970
Alg. #3	GF(2^8)	1:01.52623510	1:37.6078610

Table 2. The average time to implementation the suggested algorithms using images

Algorithm Proposed	GF Type	Encryption Time	Decryption Time
Alg. #1	GF(P)	.0092256	8.546723
Alg. #1	GF(2^8)	2.5610500	6.895475
Alg. #2	GF(P)	.0118164	12.343458
Alg. #2	GF(2^8)	2.1670292	5.4312896
Alg. #3	GF(P)	.0862004	8.927674
Alg. #3	GF(2^8)	16: 42.8670090	24:29.0339810

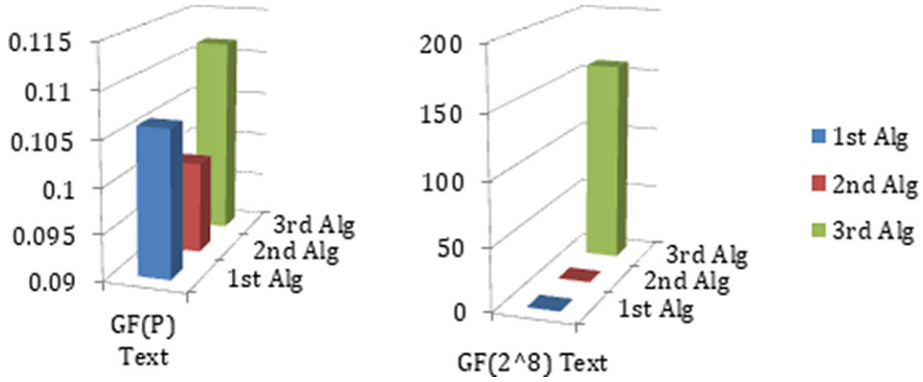

Fig. 8. Comparing between Times implementation for Algorithms suggested for texts

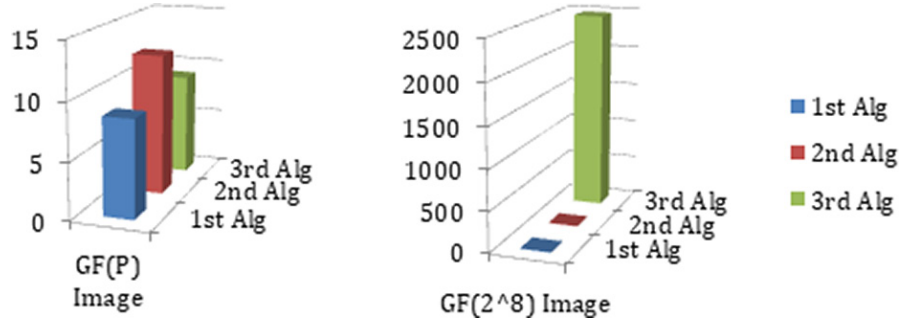

Fig. 9. Comparing between Times implementation for Algorithms suggested for images

4.4 The relationship of execution time to complexity

It is another measure that takes into account the increase in the implementation time with the increase in complexity and noting that the increase in the time of implementation was small compared to the increase in complexity or not, as detailed in Figure 10.

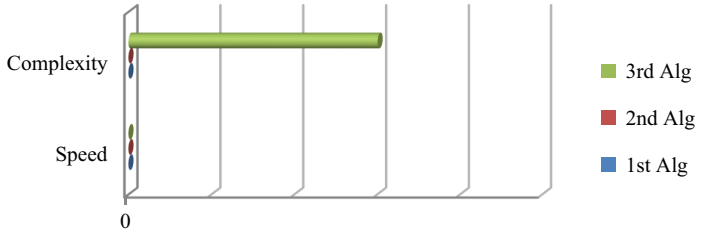

Fig. 10. Relationship between Time and complexity for Algorithms suggested

4.5 NIST test and stats

The tests are conducted for the ciphertext produced from the proposed algorithms and for each type of GF, where randomness, complexity, and repetition of characters are calculated according to the NIST tests (National Institute of Standards and Technology), as the results are shown in the Tables 3–5.

Table 3. Some of the NIST tests for Alg.1 Suggested

Statistical Measures	Frequency Test	Cumulative Sums Test	Runs Test	Longest Runs of Ones Test	Approximate Entropy Test	Serial Test
GF(P)	.571608	.095430	.794376	.163665	.823651	.486752
GF(2^8)	.169131	.183014	.767355	.613053	.874785	.637628
Result	Success	Success	Success	Success	Success	Success

Table 4. Some of the NIST tests for Alg.2 Suggested

Statistical Measures	Frequency Test	Cumulative Sums Test	Runs Test	Longest Runs of Ones Test	Approximate Entropy Test	Serial Test
GF(P)	.010909	.021819	.323143	.145110	.023640	.227638
GF(2^8)	.887537	.405915	.480335	.331698	.691367	.557825
Result	Success	Success	Success	Success	Success	Success

Table 5. Some of the NIST tests for Alg.3 are Suggested

Statistical Measures	Frequency Test	Cumulative Sums Test	Runs Test	Longest Runs of Ones Test	Approximate Entropy Test	Serial Test
GF(P)	.479500	.638440	.242142	.040080	.392533	.042004
GF(2^8)	.257899	.032419	.365342	.094576	.304355	.923116
Result	Success	Success	Success	Success	Success	Success

4.6 Histogram for images

It is another measure of the efficiency of the proposed algorithms in terms of ciphered images. Measurements were made for different types of images and each type of GF. This type of measurement is knowing the color distribution in the formed cipher and compared with the original image as shown below in Figures 11 & 12.

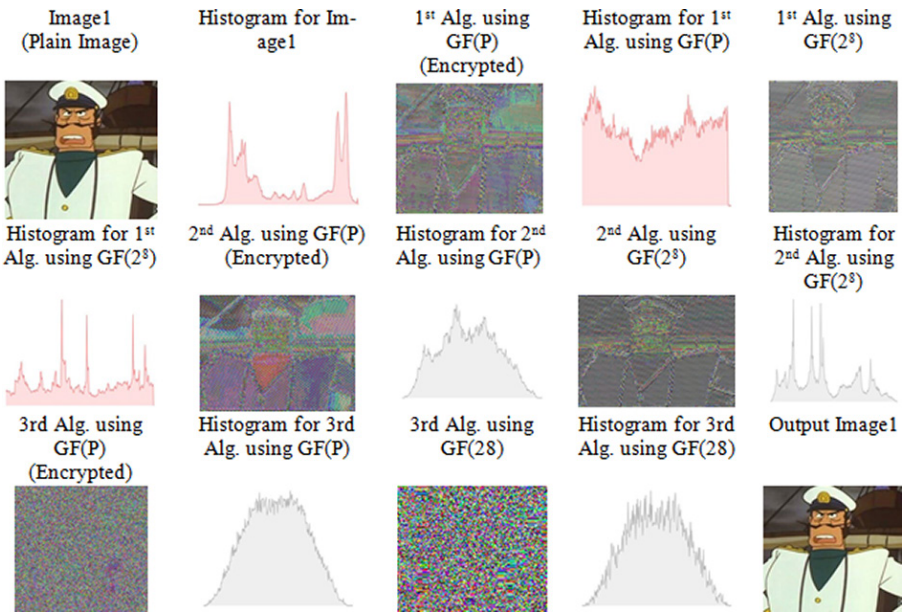

Fig. 11. Apply all the suggested methods to Image1 and find a histogram of the resulting images

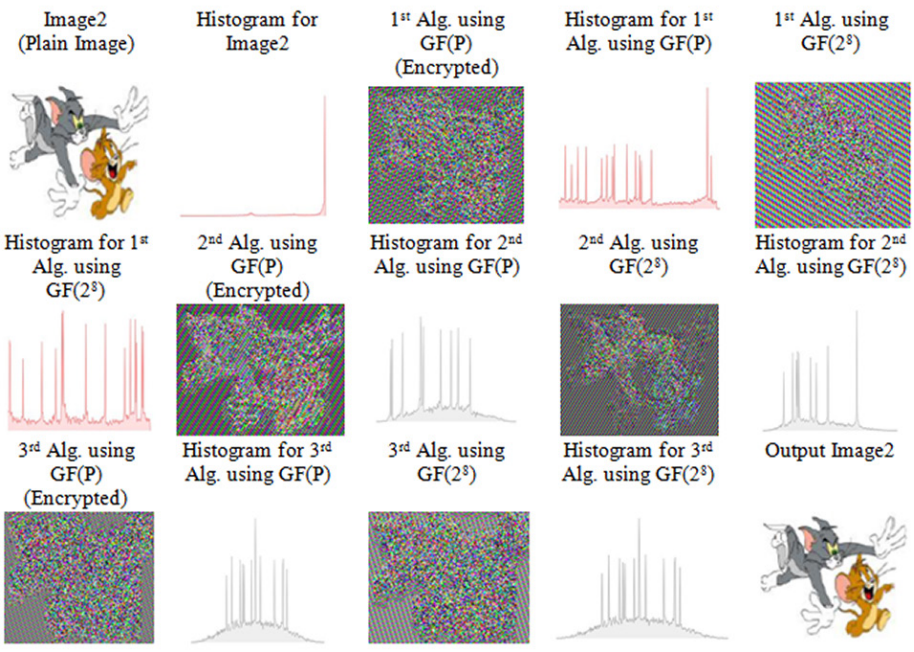

Fig. 12. Apply all the suggested methods to Image2 and find the histogram of the resulting images

5 Discussion the results

When comparing the previous results, we note that encryption using MS5 is much better in terms of complexity, speed, and randomness of the encrypted text than MS3.

As for the proposed algorithms, the first algorithm gives very good results and it is considered a successful development where the increase in the equations leads to an increase in speed in implementations. As for the second algorithm, the complexity is three times that of the first algorithm and the speed is almost equal. Whereas, if the choice is made on the number of fewer equations, it will correspondingly increase in complexity, else if the choice is on the number of more equations, it will be an increase in the speed of implementation. While the third algorithm has very high complexity compared to the previous algorithms. But it takes some time to implement, especially in pictures because it uses GF(2^8), Nevertheless, the increase in complexity remains very large against the increase in time and is considered a very excellent suggestion due to the large increase in complexity. The added complexity of an additional key adds much more complexity than the one present in MS5 because the entire additive matrix represents a key. It is preferable to encrypt text with the third algorithm, as it has very high complexity. As for images, it is preferable to encrypt them using the second algorithm, as it has good complexity and appropriate speed, and if more complexity is required, the third algorithm is the most appropriate.

6 Conclusions

The proposed algorithms provided an example of the fact that using more equations results in increasing the speed of execution. When using multi-level keys, it gives high complexity and maintains the same speed approximately. When using a high value of P in GF(P) it results in giving higher complexity with very little overtime. 8-bit encryption was used because the devices rely on 8-bit coding, and this thing is clear when dealing with images. When multiplying MS5 by a key matrix, it provides a very high complexity, and the complexity increases with increasing the value of P. On the other hand, it requires additional time in implementation, but the percentage increase of complexity is higher than the percentage increase of time. When using GF(2^8) the complexity will be multiplied by 30, because there are 30 possibilities for selecting the irreducible polynomial number in the third algorithm (multiplication operation).

7 References

[1] A. S. Rahma and Q. M. Hussein, "A New Attack on NTRU Public Key Cryptosystem Depend on Using Public Key and Public Information", Eng. & Tech. Journal, vol. 28, no. 6, 2010.

[2] G. Y. Siang, F. W. Heng, and N. H. Sarmin, "Properties and Solutions of Magic Squares", Menemui Matematik (Discovering Mathematics), vol. 34, no. 1: 63–76, 2012.

[3] D. A. Jabbar and A. S. Rahma, "Development cryptography protocol based on Magic Square and Linear Algebra System", Journal of AL-Qadisiyah for computer science and mathematics, vol. 11, no. 1, ISSN (Print): 2074–0204, ISSN (Online): 2521–3504, 2019. https://doi.org/10.29304/jqcm.2019.11.1.470

[4] I. M. Alattar and A. S. Rahma, "New Cryptography Algorithm Based On Magic Square Order Five for GF(P) and GF(28) Data", Journal of Physics: Conference Series, Publisher: IOP Publishing, ISSN: 1742–6588 (print); 1742–6596 (web), (under publication).

[5] O. A. Dawood, A. S. Rahma, and A. J. Abdul Hossen, "New Variant of Public Key Based on Diffie-Hellman with Magic Cube of Six-Dimensions", (IJCSIS), vol. 13, no. 10, October 2015.

[6] Z. Duan, J. Liu, J. Li, and C. Tian, "Improved even order magic square construction algorithms and their applications in multi-user shared electronic accounts", Theoretical Computer Science, vol. 607, no. 2015: 391–410, 2015. https://doi.org/10.1016/j.tcs.2015.07.053

[7] O. A. Dawood, A. S. Rahma, and A. J. Abdul Hossen, "Generalized Method for Constructing Magic Cube by Folded Magic Squares", I.J. Intelligent Systems and Applications, 2016, 1, 1–8, Published Online in MECS (http://www.mecs-press.org/), DOI: https://doi.org/10.5815/ijisa.2016.01.01

[8] S. U. Umar, "An Improved RSA based on Double Even Magic Square of order 32", Kirkuk University Journal /Scientific Studies (KUJSS), vol. 12, no. 4, September 2017, ISSN 1992–0849, 2017. https://doi.org/10.32894/kujss.2017.132388

[9] R. H. AL-Hashemy and S. A. Mehdi, "A New Algorithm Based on Magic Square and a Novel Chaotic System for Image Encryption", J. Intell. Syst. vol. 29, no. 1: 1202–1215, 2020. https://doi.org/10.1515/jisys-2018-0404, Received October 3, 2018; previously published online, February 1, 2019. https://doi.org/10.1515/jisys-2018-0404

[10] S. M. Kareem and A. S. Rahma, "A novel approach for the development of the Twofish algorithm based on multi-level key space", Journal of Information Security and Applications, Journal of Information Security and Applications 50 - 102410, 2020. https://doi.org/10.1016/j.jisa.2019.102410

[11] G. D. Cuevas and T. Netzer, "Quantum information theory and free semialgebraic geometry:one wonderland through two looking glasses", arXiv:2102.04240v1 [quant-ph] (8 Feb 2021).

[12] O. A. Dawood, A. S. Rahma, and A. J. Abdul Hossen, "Public Key Cipher with Signature Based on Diffie-Hellman and the Magic Square Problem", Eng. & Tech. Journal, vol. 34, Part (B), no. 1, 2016.

[13] S. M. Kareem and A. S. Rahma, "A Modification on Key Stream Generator for RC4 Algorithm", Engineering and Technology Journal, vol. 38, Part B (2020), no. 20, pp. 54–60, 2020. https://doi.org/10.30684/etj.v38i2B.404

[14] S. M. Kareem and A. S. Rahma, "New modification on feistel DES algorithm based on multi-level keys", International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 3, June 2020, pp. 3125–3135, ISSN: 2088–8708, DOI: https://doi.org/10.11591/ijece.v10i3.pp3125-3135, 2019.

[15] S. M. Kareem and Abdul Monem S. Rahma, "A new multi-level key block cypher based on the Blowfish algorithm", TELKOMNIKA Telecommunication, Computing, Electronics and Control, vol. 18, no. 2, April 2020, pp. 685–694, ISSN: 1693–6930, accredited First Grade by Kemenristekdikti, Decree no: 21/E/KPT/2018, DOI: https://doi.org/10.12928/telkomnika.v18i2.13556, 2020.

[16] M. Al-dabag, H. S. ALRikabi, and R. Al-Nima, "Anticipating Atrial Fibrillation Signal Using Efficient Algorithm," International Journal of Online and Biomedical Engineering (iJOE), vol. 17, no. 2, pp. 106–120, 2021 https://doi.org/10.3991/ijoe.v17i02.19183

[17] A. S. Hussein, R. S. Khairy, S. M. M. Najeeb, and H. T. Salim, "Credit Card Fraud Detection Using Fuzzy Rough Nearest Neighbor and Sequential Minimal Optimization with Logistic Regression," International Journal of Interactive Mobile Technologies, vol. 15, no. 5, pp. 24–42, 2021. https://doi.org/10.3991/ijim.v15i05.17173

[18] H. Naman, N. Hussien, M. Al-dabag, and Haider Th.Salim Alrikabi, "Encryption System for Hiding Information Based on Internet of Things," International Journal of Interactive Mobile Technologies, vol. 15, no. 2, pp. 172–183, 2021. https://doi.org/10.3991/ijim.v15i02.19869

8 Authors

Ibrahim Malik Abdul Rahman Al-Attar was born in Baghdad (1994), obtained a bachelor's degree in computer science – data security from the University of Technology – Baghdad (2016), currently a master's in computer science/data security – research phase at the University of Technology – Baghdad. I previously published many papers in local and international journals, and I still have some papers in progress. I have programmed some electronic systems in companies. Housing: Iraq – Baghdad – Eastern Karrada. Email: ibrahiminter@yahoo.com

Prof. Rahma has an extensive background in Cryptography and Information Security Image Processing, Pattern Recognition, and Biometrics. He received his Ph.D. in Computer Science in 1984, from the Loughborough University of Technology in the United Kingdom, and has become a professor in Computer Science since 2008. His main work experience involves teaching at Iraqi universities and supervising postgraduate students; He also was Deputy Dean of the Department of Computer Science, University of Technology, Iraq from 2005 to 2013. From 2013 to 2015, he became the Dean of the department. Now He is a Lecturer and the head of the Department of Computer Engineering Techniques, Imam Ja'afar Al-Sadiq University. Prof. Rahma published 180 Papers, 4 Books in Computer Science; supervised 36 Ph.D. and 66 M.Sc. students. Email: 110003@uotechnology.edu.iq

Article submitted 2021-05-03. Resubmitted 2021-06-02. Final acceptance 2021-06-04. Final version published as submitted by the authors.