Al-Qadisiyah Journal of Pure Science

Volume 26 | Number 4 Article 53

8-15-2021

A Comparison Between Odd Magic Squares Use In Cryptographic Algorithms

Ibrahim Malik Alatta
Department of Computer Science, University of Technology, Baghdad, Iraq, ibrahiminter@yahoo.com

Abdul Monem S. Rahma
Department of Computer Science, University of Technology, Baghdad, Iraq, 110003@uotechnology.edu.iq

Recommended Citation
Alatta, Ibrahim Malik and Rahma, Abdul Monem S. (2021) "A Comparison Between Odd Magic Squares Use In Cryptographic Algorithms," *Al-Qadisiyah Journal of Pure Science*: Vol. 26: No. 4, Article 53.
DOI: 10.29350/qjps.2021.26.4.1399
Available at: https://qjps.researchcommons.org/home/vol26/iss4/53

Al-Qadisiyah Journal of Pure Science
ISSN(Printed): 1997-2490 ISSN(Online): 2411-3514
DOI: /10.29350/jops.
http://qu.edu.iq/journalsc/index.php/JOPS

A Comparison between odd magic squares use in cryptographic algorithms

Authors Names
a.Ibrahim Malik ALatta
b.Abdul Monem S. Rahma

Article History
Received on:25/6/2021
Revised on: 25/8/2021
Accepted on:30/8/2021
Keywords:
Odd Magic Square,
Cryptography,
Linear equation system,
Finite field,
GF(2^8).

DOI:h https://doi.org/10.29350/jops. 2021.26. 4.1399

ABSTRACT

This paper has been developed to compare encryption algorithms based on individual magic squares and discuss the advantages and disadvantages of each algorithm or method. Where some positions of the magic square are assigned to the key and the remaining positions are assigned to the message, then the rows, columns and diagonals are summed and these results are as ciphertext and in the process of decryption the equations are arranged and solved by Gauss elimination metod. All algorithms were applied to encrypte the text and images, as well as using both GF(P) and GF(2^8), and the speed and complexity were calculated. The speed of MS9 by using GF(P) is 15.09085 Millie Second, while by using GF(2^8) it will be 18.94268 Millie Second, and the complexity is the value of the ASCII code raised to the exponent of the number of message locations multiplied by the value of the prime number raised to the exponent of the number of key locations.

1. Introduction

Due to the development of software in the modern age of life, the need to develop encryption algorithms to protect data when transmitted over the Internet [1].

Which led to the development of many different encryption algorithms, and the focus was on developing encryption algorithms that are not known (known algorithms as AES,RSA,DES,...etc) in general and focusing on their development [2].

In this paper, the advantages and disadvantages of encryption algorithms using several types of single magic squares of different sizes compared [3].

[a, b] Department of Computer Science, University of Technology, Baghdad, Iraq.

Email: [a]ibrahiminter@yahoo.com [b]110003@uotechnology.edu.iq

Since the idea of the emergence of the magic square is very old, dating back to B.C times, as it was found in ancient books such as one of the books of one of the well-known chemists, Jabir bin Hayyan [4].

As well as entered magic squares in several different fields such as sorcery, astronomy and many games [5].

Mathematicians and cryptologists are the other ones who used magic squares and it was introduced into the field of cryptography [6].

A group of intelligence games whose idea was based on the idea of magic squares such as chess, Sudoku and others, as the movement of the clicker in the chess game was the basis of its idea based on the magic square as mentioned by some books [5].

The following is a set of previous works related to the idea of the current paper, which based on the cryptography or the magic square:

In 2009, Ganapathy et al developed an encryption algorithm that produces a different ciphertext using the magic square as an alternative approach to dealing with ASCii code , which depends on the seed number of the magic square and the starting number so that the resulting result cannot be traced easily, which gives high security [7].

In 2014, Dharini et al. developed an encryption technology to increase security in cloud computing in a way that the use of the RSA algorithm was combined with the magic square to provide more security [8].

In 2015, Duan et al. developed an encryption algorithm based on the idea of a odd magic square using two specific magic squares, and the complexity and speed of the proposed work were calculated [9].

In 2017, Umar proposed an encryption algorithm to get rid of the problem of repetition in the ASCii code using the magic square of order 32, which was able to easily track the matter and solve the problem and give more security and provide a high level of security [10].

In 2019, AL-Hashemy et al proposed a 3D encryption algorithm to encrypt color images based on the magic square, where a number of random keys are generated in the size of the image to be encrypted, then the image is divided and the magic square is used, then the XOR binary operation is used [11].

In 2020, Mohammed et al proposed an encryption algorithm based on the magic square and with the help of matrix with size 4×4 , where the proposed algorithms were used for encryption and data hiding, relying mainly on the magic square, and all results depended on the specified field $GF(2^8)$ [12].

2. Previously technologies and advantages

Magic squares have many well-known distinctive properties so that, as in the game of Sudoku, the sum of each row, column or diagonal is one fixed result that called the magic constant. This kind of magic square called a normal magic square [13].

The magic square called Prime Magic Square if it contains all the properties of the normal magic squares in addition to the fact that all the numbers in the magic square are prime numbers [13].

The magic square is treated like ordinary matrices, as it has different sizes such as matrices where it is called magic square of order three (MS3), magic square of order four (MS4), magic square of order five (MS5)....etc [3].

The magic square is called the composite Magic Square if it contains all the properties of the normal magic squares, in addition to that if the outer numbers frame of the magic square is removed, the remaining result will remain a magic square and so on until reaching to the smallest magic square size [4].

Magic squares are used in encryption, where some locations are assigned to the key and the remaining locations are assigned to the message. Then the sum of each row, column and diagonal is calculated according to the known properties of the magic square [14].

Contrary to the properties of the magic square that the sum of the row, column or diameter is equal, in this case of using the encryption algorithms , the sums are not required to be equal [14].

After studying these equations resulting from summing all the rows, columns and the main and secondary diagonals every one of them separately, it was found that there is a dependency in two of them in every magic square, and this reliability includes all the odd magic squares that are being studied in this paper [14].

In all the magic squares that have been studied in this paper, the first cancelled equation is located in the middle of the rows and the second in the middle of the columns in relation to the magic squares of order three (MS3), of order five (MS5), of order seven (MS7) and of order nine (MS9) as in the following figure 1[15]:

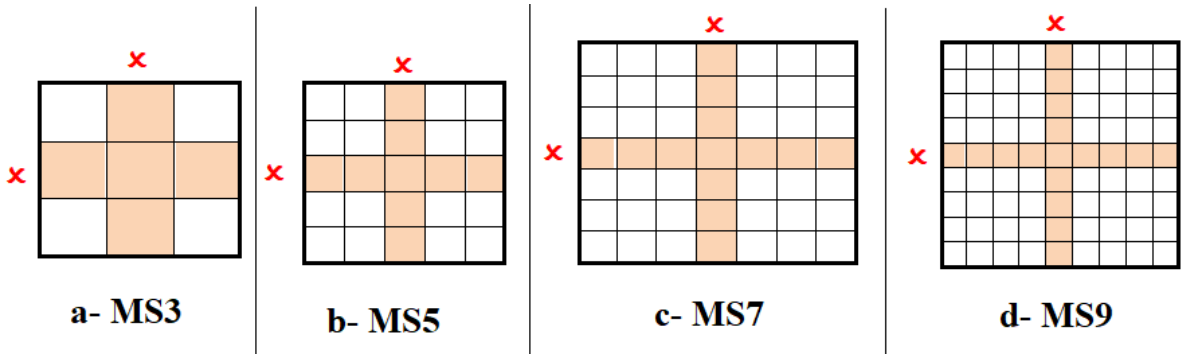

Figure 1. The cancelled equations in the odd magic square.

These resulting sums represent ciphertexts that the recipient will receive. Which its number is equal to the number of the message locations in the magic square used [15].

On the other hand, the recipient will receive the encrypted texts with his possession of the key as value and positions and will solve the equations together as linear equations to get the original information sent, whether texts or images [15].

The proposed system for solving equations is all subject to the specified field, which depends on a certain prime number [5].

Through the analyzes and statistics used in this paper, the equations have been solved by Gaussian elimination [16].

Since the current devices used depend on the 8-bit system, the proposed system was developed with encryption from the field based on the prime number GF(P) to the use of the polynomial number, which all field is based on GF(2^8) [6].

These techniques in the magic squares depend on generating keys and assigning randomly generated and their values are between (0-255) [16].

As for the distribution of key locations, it is done randomly, provided that a line, row, or column is not completely canceled [15].

3. A set of cipher examples using odd magic squares:

Here are full examples showing the encryption and decryption processes for odd magic squares, as follows:

3.1 Example of using MS3:

- **The Encryption**

  It is assumed that the key values used are as { 3 , 7 , 9}.

  While the values of the message to be encrypted are { 4 , 4 , 4, 4 , 4 , 4}.

  It is assumed that the following key locations have been chosen:

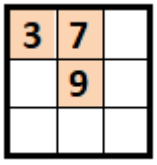

Figure 2. The key choosing in MS3.

Then the remaining positions are filled with the message as follows :

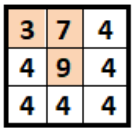

Figure 3. The MS3 with key and message.

Then the six totals are calculated according to the following equations:

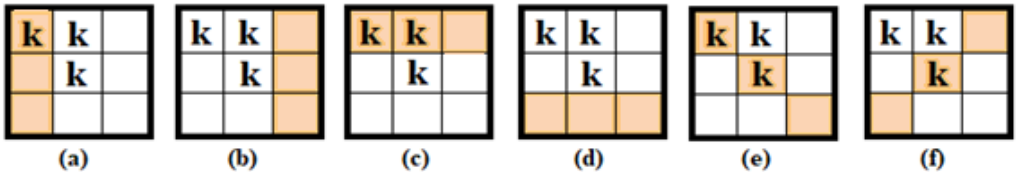

Figure 4. The equations used in MS3.

Thus, the following sums will be formed, which will be a ciphertext sent to the other party.

Sum1 = 11 , Sum2 = 12 , Sum3 = 14 , Sum4 = 12 , Sum5 = 16 , Sum6 = 17.

- **The Decryption**

The receiving party will have the key and the ciphertext and have 6 unknowns whose values will be the values of the sent message when known. The unknown values will be assumed by X1 - X6 as follows:

3	7	K1
K2	9	K3
K4	K5	K6

Figure 5. Arranging the known information by the recipient prior to the solution.

And based on the six equations that he adopted in the encryption process, the following table will be formed by the recipient. So that if the unknown is present in the equation, the value 1 will be placed in the table position, and if it is not present, the key number will be present, the value 0 will be placed in the table position and the key value will be subtracted from the old sum.

X_1	X_2	X_3	X_4	X_5	X_6	rest of the sum (old sum – the key value if found)
0	1	0	1	0	0	8
1	0	1	0	0	1	12
1	0	0	0	0	0	4
0	0	0	1	1	1	12
0	0	0	0	0	1	4
1	0	0	1	0	0	8

The equations will be arranged so that the principal diagonal does not contain a zero as follows:

X_1	X_2	X_3	X_4	X_5	X_6	rest of the sum (old sum – the key value if found)
1	0	0	0	0	0	4
0	1	0	1	0	0	8
1	0	1	0	0	1	12
1	0	0	1	0	0	8
0	0	0	1	1	1	12
0	0	0	0	0	1	4

The previous equations will be solved together by Gaussian elimination, where the triangle part below the main diagonal will be solved first, where all numbers outside the main diagonal will be eliminated by using the Gaussian elimination rules which represent by adding or subtracting two equations together or multiplying a certain equation by a constant and depending on GF(P), then the triangle that lies above the main diagonal is solved also by applying the Gaussian rules of elimination down to the following form:

X_1	X_2	X_3	X_4	X_5	X_6	rest of the sum (old sum – the key value if found)
1	0	0	0	0	0	4
0	1	0	0	0	0	4
0	0	1	0	0	0	4
0	0	0	1	0	0	4
0	0	0	0	1	0	4
0	0	0	0	0	1	4

Then the resulting final sums will represent the value of the original texts sent

{ 4 , 4 , 4 , 4 , 4 , 4}.

3.2 Example of using MS5:

- **The Encryption**

By using MS5 there will be 5 sums for rows and 5 sums for columns with two sums for each diagonal , the grand total will be 12 sums, and that there are two sums of them that will cancel out due to the dependencies which are the row in the middle and the column in the middle, so the sums that will be used in the encryption are 10 sums.

Thus it is assumed that the selected key locations are as follows:

	k	k		
k		k	k	
k	k		k	k
k	k			k
	k	k	k	

Figure 6. The key choosing in MS5.

It is assumed that the message to be encrypted is {5,5,5,5,5,5,5,5,5,5}

And the key used is {1, 2, 3, 4, 6, 1, 2, 3, 4, 6, 1, 2, 3, 4, 6 }.

MS5 will be as follow:

5	1	2	5	5
3	5	4	6	5
1	2	5	3	4
6	1	5	5	2
5	3	4	6	5

Figure 7. MS5 before the encryption.

The ten sums will be calculated and the results will be as follows, and they will be considered ciphertext and sent to the second party.

Sum1=18 , sum2=23 , sum3=19 , sum4=23 , sum5=20 ,sum6=12 , sum7=25 , sum8 = 21 , sum9=25 , sum10=22.

- **The Decryption**

The receiving party will have the ciphertext and the following information where it will enforce the unknowns X1 -X10.

K1	1	2	K2	K3
3	K4	4	6	K5
1	2	K6	3	4
6	1	K7	K8	2
K9	3	4	6	K10

Figure 8. The arranging information by The recipient for MS5.

By creating the solution table, putting the value 1 in the case of found the unknown in the equation, putting zero in the case of found the key value and subtracting its value from the sum, the solution will be as follows:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	SUM
1	1	1	0	0	0	0	0	0	0	15
0	0	0	1	1	0	0	0	0	0	10
0	0	0	0	0	0	1	1	0	0	10
0	0	0	0	0	0	0	0	1	1	10
1	0	0	0	0	0	0	0	1	0	10
0	0	0	1	0	0	0	0	0	0	5
0	1	0	0	0	0	0	1	0	0	10
0	0	1	0	1	0	0	0	0	1	15
1	0	0	1	0	1	0	1	0	1	25
0	0	1	0	0	1	0	0	1	0	15

After arranging the equations, we get the following:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	sums
1	0	0	1	0	1	0	1	0	1	25
1	1	1	0	0	0	0	0	0	0	15
0	0	1	0	1	0	0	0	0	1	15
0	0	0	1	0	0	0	0	0	0	5
0	0	0	1	1	0	0	0	0	0	10
0	0	1	0	0	1	0	0	1	0	15
0	0	0	0	0	0	1	1	0	0	10
0	1	0	0	0	0	0	1	0	0	10
1	0	0	0	0	0	0	0	1	0	10
0	0	0	0	0	0	0	0	1	1	10

After applying the Gaussian elimination rules based on GF(P), we will get the original text in the sum column as follows:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	sums
1	0	0	0	0	0	0	0	0	0	5
0	1	0	0	0	0	0	0	0	0	5
0	0	1	0	0	0	0	0	0	0	5
0	0	0	1	0	0	0	0	0	0	5
0	0	0	0	1	0	0	0	0	0	5
0	0	0	0	0	1	0	0	0	0	5
0	0	0	0	0	0	1	0	0	0	5
0	0	0	0	0	0	0	1	0	0	5
0	0	0	0	0	0	0	0	1	0	5
0	0	0	0	0	0	0	0	0	1	5

Also, a text message encrypted and decrypted was done with MS5 and a clarification of the Gaussian elimination process see appendix 1.

3.3 Example of using MS7:

- **The Encryption**

In the case of MS7 there will be 14 sums as 7 sums for rows, 7 sums for columns, and 2 sums for each of the diagonals, the cancel two sums for each row and column (which are in the middle).

It is assumed that the message to be encrypted is {3,3,3,3,3,3,3,3,3,3,3,3,3,3} and the key used is

{2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,…….70} (the length of the key is 35).

It assumed that the selected key locations are as follows:

	k	k	k	k		
k		k	k	k	k	k
k	k	k	k	k		k
k				k	k	
k		k	k		k	k
k	k		k	k	k	k
	k	k	k	k	k	

Figure 9. The key positions in MS7.

After placing the key values in their positions in MS7 and filling the remaining locations with the message, 14 equations will be collected and the result will be sent as ciphertext to the other party.

- **The Decryption**

After the keys are placed in the agreed positions as in Figure 9 and the remaining locations will remain unknown the unknowns will be X1-X35.

Then the table will be formed as in the example of using MS3 and MS5 by putting the number 1 in the formed table in the case of found the unknown in the equation used and putting the value zero if there is a value for the key and subtracting its value from the sum.

Then the equations are arranged so that the main diagonal does not contain the value zero.

Then The Gaussian elimination rules are used based on GF(P) to retrieve the sent message {3,3,3,3,3,3,3,3,3,3,3,3,3,3} .

3.4 Example of using MS9:

- **The Encryption**

When using MS9, there will be 18 sums after cancelled which have dependencies.

And it is assumed that the following key locations have been selected:

K	K	K	K	K	K		K	
K		K	K	K	K	K		K
K	K	K	K	K		K	K	K
K		K	K	K	K	K	K	K
	K	K			K	K	K	K
K		K	K	K	K		K	K
	K	K	K	K		K	K	K
K	K		K	K		K		K
K	K	K	K	K	K		K	

Figure 10. The key positions in MS9.

Where there will be 63 key locations. The 18 sums will be calculated and the results will be sent as

encrypted text to the other party.

- **The Decryption**

As in MS3 , MS5 , and MS9 a table will be created based on key values and ciphertext , values for unknowns X1-X18 will be assumed.

Then the equations will be arranged. Then the equations will be solved by Gaussian elimination and depending on GF(P) to retrieve the original text sent.

4. Analysis of the results:

4.1 Comparison Summary

After applying the previous proposed algorithms MS3, MS5, MS7 and MS9, and by using each of GF(P) and GF(2^8) ,the value of speed, complexity and other information has been put in a table to summarize everything that matters to the algorithms and give a brief and accurate summary as follows in Table 1.

Table1. Comparison of the sizes of different magic squares.

MS Order	The type of GF	Message positions	keys positions	The complexity	Encryption / Decryption average time
MS3	GF(P)	6	3	$(256)^6 \times (P)^3$	27.45215
MS3	GF(2^8)	6	3	$(256)^6 \times (256)^3$	07.89478
MS5	GF(P)	10	15	$(256)^{10} \times (P)^{15}$	24.54793
MS5	GF(2^8)	10	15	$(256)^{10} \times (256)^{15}$	11.79402
MS7	GF(P)	14	35	$(256)^{14} \times (P)^{35}$	18.14987
MS7	GF(2^8)	14	35	$(256)^{14} \times (256)^{35}$	16.20036
MS9	GF(P)	18	63	$(256)^{18} \times (P)^{63}$	15.09085
MS9	GF(2^8)	18	63	$(256)^{18} \times (256)^{63}$	18.94268

The following Figures 12 and 11 show a comparison of the complexity of magic squares using GF(P) and GF(2^8), respectively.

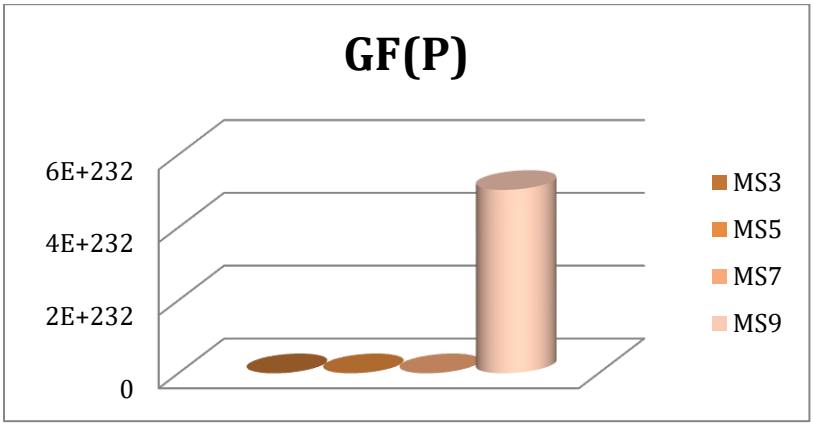

Figure 11. The complexity in the different sizes of MS using GF(P).

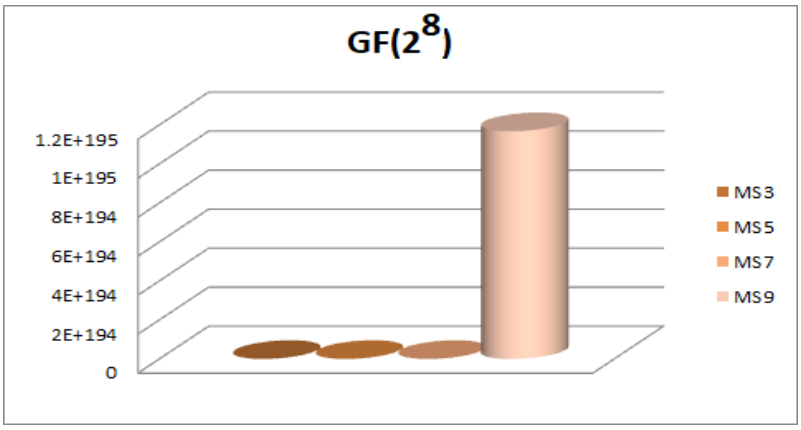

Figure 12. The complexity in the different sizes of MS using GF(2^8).

The following Figures 13 and 14 show a comparison of the execution time of magic squares using GF(P) and GF(2^8), respectively.

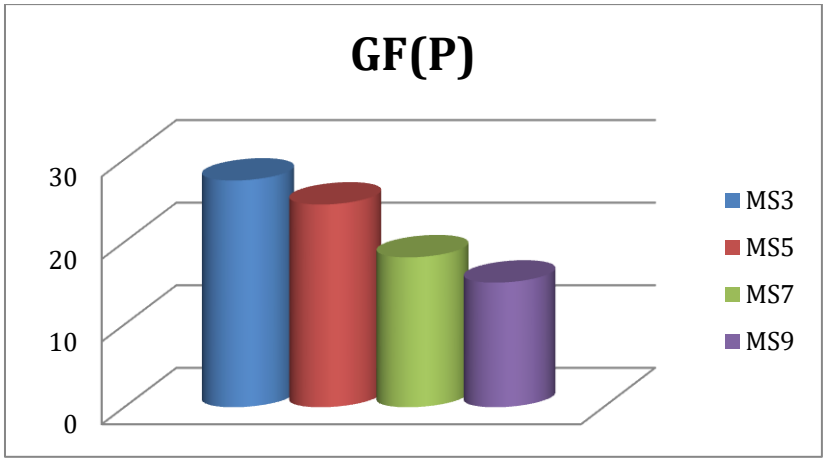

Figure 13. The execution time in the different sizes of MS using GF(P).

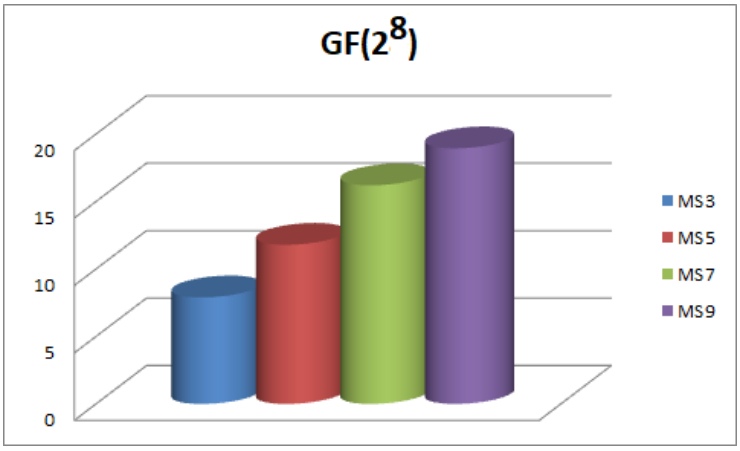

Figure 14. The execution time in the different sizes of MS using GF(2^8).

4.2 Discuss the results

We notice from Table No. 1 that if the size of the magic square increases, the complexity will increase dramatically, especially when the key locations increase.

We also note that using GF(P) if the size of the magic square increases, the execution time will decrease.

While using GF(28) if the size of the magic square increases, the execution time decreases.

And when consider the rate of increase in complexity with the increase in time, will note that the increase in complexity is very large compared to the increase in execution time.

Therefore, we can consider that MS9 is better than MS7, which in turn is better than MS5, and MS5 is better than MS3 in encryption.

5. Conclusions

When comparing the different sizes of magic squares, many conclusions were obtained from them: If a large prime number is used, the complexity will be high as well.

When the number of message locations in the magic square increases, the speed will increase.

The larger the size of the magic square will give better results, meaning that MS9 is better than the rest of the magic squares that were compared.

When using GF(P) on the larger magic squares it will be faster than if it is used on the smaller magic squares.

References

[1] I. M. Alattar and A. S. Rahma , " A block cipher algorithm developed using Magic Square in the seventh order", 2nd international virtual conference on pure science IOP publishing , Journal of Physics : Conference Series , 1999, 012119, 2021.

[2] I. M. Alattar and A. S. Rahma , " New Cryptography Algorithm Based On Magic Square Order Five for GF(p) and GF(2^8) Data" , IOP publishing , Journal of Physics: Conference Series (under publishing).

[3] 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. 29(1): 1202–1215, 2019.

[4] S. Cichacz and T. Hincbc , " A magic rectangle set on Abelian groups and its application" , Discrete Applied Mathematics, Volume 288, Pages 201-210, 2021.

[5] A. Dharini, R. M. 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 Nov. 2014, pp. 439-444 2014.

[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 – Elsevier , 2015.

[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.

[8] G. Ganapathy, and K. Mani, " Add-On Security Model for Public-Key Cryptosystem Based on Magic Square Implementation " , Proceedings of the World Congress on Engineering and

Computer Science 2009 Vol I ,WCECS 2009, San Francisco, USA,2009.
[9] D. A. Jabbar and A. S. Rahma ," Proposed Cryptography Protocol based on Magic Square, Linear Algebra System and Finite Field ", Jour of Adv Research in Dynamical & Control Systems, Vol. 10, No. 10 , 2018.
[10] D. A. Jabbar and A. S. Rahma , " Development cryptography protocol based on Magic Square and Linear Algebra System" , Vol.11 No.1 2019.
[11] S. M. Kareem and A. S. Rahma , " A Modified On Twofish Algorithm Based On Cyclic Group And Irreducible Polynomial In Gf (28) , Al-Qadisiyah Journal Of Pure Science (QJPS) , Vol. 25, No. 1, pp Comp. 1 –9, 2020.
[12] 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.
[13] V. Nandalal and V. Anand Kumar , " Design and Analysis of (5, 10) Regular LDPC Encoder Using MRP Technique" , Wireless Personal Communications volume 118, pages1295–1311 , 2021.
[14] Stallings, William, “Cryptography and network security: principles and practice 6 Edition,” Person Education Inc. 2014.
[15] S. U. Umar , " An Improved RSA based on Double Even Magic Square of order 32", Kirkuk University Journal /Scientific Studies (KUJSS), Volume 12, Issue 4, 2017.
[16] A. Zadeh, S. Benoit and L. Morency , "StarNet: Gradient-free Training of Deep Generative Models using Determined System of Linear Equations" , arXiv Preprint. Ongoing research at CMU , arXiv:2101.00574v1 [cs.LG] ,2021.

Appendix 1

Will encrypt the message (accountant) by using MS5 and the prime number chosen is (251) and the key value is all 10 and it's locations will chose randomly as follows

		10	10	
10			10	10
10	10		10	10
10	10	10		
	10	10	10	

The remaining locations will filled with the message after convert it's letters into ASCII code as followed :

a = 97, c=99, o=111, u=117, n=110, t=116.

Then the MS5 will be as follow:

97	99	10	10	99
10	111	117	10	10
10	10	110	10	10
10	10	10	116	97
110	10	10	10	116

The encryption process includes calculating the sums for rows, columns, and diagonals, and deleting what causes dependency, so there are 10 sums as follows: {sum1= 315 mod P = mod 251 = 64, sum2 = 7, sum3 = 243, sum4 = 5, sum5 = 237, sum6 = 240, sum7 = 156 , sum8 = 80, sum9 = 48, sum10 = 88).

The decryption process done by put the value 1 if found the letter in the equation or put 0 if not found and the remainder will be the sum received minus the values of keys in that equation as shown:

a	c	c	o	u	n	t	a	n	t	rem
1	1	1	0	0	0	0	0	0	0	44
0	0	0	1	1	0	0	0	0	0	228
0	0	0	0	0	0	1	1	0	0	213
0	0	0	0	0	0	0	0	1	1	226
1	0	0	0	0	0	0	0	1	0	207
0	1	0	1	0	0	0	0	0	0	210
0	0	0	0	0	0	1	0	0	0	116
0	0	1	0	0	0	0	1	0	1	61
1	0	0	1	0	1	1	0	0	1	48
0	0	1	0	0	1	0	0	1	0	68

Then reordering the rows as the main diagonal not contains 0 as followed:

a	c	c	o	u	n	t	a	n	t	rem
1	0	0	0	0	0	0	0	1	0	207
1	1	1	0	0	0	0	0	0	0	44
0	0	1	0	0	0	0	1	0	1	61
0	1	0	1	0	0	0	0	0	0	210
0	0	0	1	1	0	0	0	0	0	228
0	0	1	0	0	1	0	0	1	0	68
0	0	0	0	0	0	1	0	0	0	116
0	0	0	0	0	0	1	1	0	0	213
0	0	0	0	0	0	0	0	1	1	226
1	0	0	1	0	1	1	0	0	1	48

Then these equations will be solved by the gauss eliminations first will solve the down triangle after that will be solve the up triangle, in the first triangle will be start from the first column up to down then the second column ….etc.

The process of the solution include sum or subtract the rows or multiply number by row to get rid of numbers and make all zeros as shown:

Row2 = (Row2 – Row1) mod 251 as shown:

0	1	1	0	0	0	0	0	250	0	88

Then the number 1 at the end of the first column is eliminated by:

Row10 = (Row10 – Row1) mod 251.

And continue for all the down rectangle until reach to these steps:

a	c	c	o	u	n	t	a	n	t	rem
1	0	0	0	0	0	0	0	1	0	207
0	1	1	0	0	0	0	0	250	0	88
0	0	1	0	0	0	0	1	0	1	61
0	0	0	1	0	0	0	1	1	1	183
0	0	0	0	1	0	0	250	250	250	45
0	0	0	0	0	1	0	250	1	250	7
0	0	0	0	0	0	1	0	0	0	116
0	0	0	0	0	0	0	1	0	0	97
0	0	0	0	0	0	0	0	1	1	226
0	0	0	0	0	0	0	0	0	4	213

Then continue to the second rectangle as the same methods used until reaching to that:

a	c	c	o	u	n	t	a	n	t	rem
1	0	0	0	0	0	0	0	0	0	97
0	1	0	0	0	0	0	0	0	0	99
0	0	1	0	0	0	0	0	0	0	99
0	0	0	1	0	0	0	0	0	0	111
0	0	0	0	1	0	0	0	0	0	117
0	0	0	0	0	1	0	0	0	0	110
0	0	0	0	0	0	1	0	0	0	116
0	0	0	0	0	0	0	1	0	0	97
0	0	0	0	0	0	0	0	1	0	110
0	0	0	0	0	0	0	0	0	1	116

Then the reminders represent the original message before convert it from ASCII to character as follow:

a = 97, c = 99 , …etc. ➔ the message is "accountant".