International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V70I3P215

Volume 70 Issue 3, 132-139, March 2022
© 2022 Seventh Sense Research Group®

Original Article

An Enhanced Playfair Algorithm with Dynamic Matrix Using the Novel Multidimensional Element-in-Grid Sequencer (MEGS)

Jan Carlo T. Arroyo[1], Ariel M. Sison[2], Ruji P. Medina[3], Allemar Jhone P. Delima[4]

[1,3]Graduate Programs, Technological Institute of the Philippines, Quezon City, Philippines
[1]College of Computing Education, University of Mindanao, Davao City, Davao del Sur, Philippines
[1,4]Institute of Information and Computer Studies, Northern Iloilo Polytechnic State College, Iloilo, Philippines
[2]School of Computer Studies, Emilio Aguinaldo College, Manila, Philippines

[1]jancarloarroyo@nipsc.edu.ph, [2]ariel.sison@eac.edu.ph, [3]ruji.medina@tip.edu.ph

Abstract — *This study enhanced the Playfair algorithm with the novel Multidimensional Element-in-Grid Sequencer (MEGS). A 16x16 dynamic matrix with a new character sequencing scheme is introduced before substitution for a more secure encryption process. The proposed modification incorporates matrix rotation, matrix shifting, matrix rolling, and crossover operations in producing the ciphertext. The enhanced Playfair algorithm will pave the way for a robust system to secure information where similar plaintext characters may not have the same encryption value. The generated ciphertext will only contain printable ASCII characters. Simulation results revealed that the modified Playfair algorithm obtained an average of 53.54% avalanche effect when tested using plaintext with varying lengths ranging from 10 to 1000 characters, thus, surpassing the Strict Avalanche Criterion (SAC) standard. Applying the modified Playfair algorithm in image steganography or password security is recommended for future works, and other performance metrics such as the randomness test and brute force attack analysis be tested.*

Keywords — *Modified Playfair algorithm, MEGS-based Playfair algorithm, matrix rotation, matrix rolling, matrix shift.*

I. INTRODUCTION

Cryptography is the science and mathematics of hiding and obscuring data into an unintelligible format to protect adversaries [1]–[3]. Information security has gained more interest for most organisations as the electronic exchange of information attracts more attackers, hence the call for a more robust data transfer and communication security [4].

In cryptography, a cipher is an algorithm for performing encryption and decryption. Several cryptographic algorithms have been developed to transform plaintext into an incomprehensible format. These include the Playfair algorithm [5]–[7], Polybius Square [8]–[12], Caesar Cipher [13]–[15], and Rail fence Cipher [16], [17], among others. Among these, Playfair offers simplicity in operation but is robust due to its practical digraph substitution that inhibits single letter frequencies where plaintext letters are no longer identified as entities [18]. Being one of the widely used encryption algorithms [19], Playfair has been a vital component in the security of applications, such as in the encryption of patient records using a 5x5 matrix [20] and 15x14 matrix [21], in the key generation process of watermarking medical images [22], in image steganography systems using 16x16 with XOR operations [23], [24], in image encryption-compression system with the discrete wavelet transform (DWT) [25], in message encryption system with radio mean labelling method [26], and in client-side encryption for cloud security using a 9x9 matrix [27] and a 16x16 matrix [28], to name some.

More so, various modifications in the Playfair algorithm have been introduced to adjust its encryption process to enhance the security of the cipher. However, these modifications still have limited and static matrices [19], [29], [30], which in turn produces repetitive ciphertext patterns that are vulnerable to attacks. Further, other modifications incorporated non-printable characters in their matrices. Though non-printable characters may increase the complexity of the Playfair algorithm, these are difficult to display on screens and may negatively affect the ciphertext structure [31], [32], leading to corruption or loss of data [33], [34].

Furthermore, the transfer of non-printable characters is impossible in some communication lines [35]. Ciphertext with non-printable characters, such as control sequences, cannot be keyed in, hence, making decryption impossible for end-users [36]. These problems are traceable to the matrix elements and the static distribution of the elements in the matrix. Moreover, the use of static matrices compromises the algorithm's security [37] and, therefore, should be replaced with dynamic substitution tables to be more resistive against cryptanalysis attacks. [38], [39].

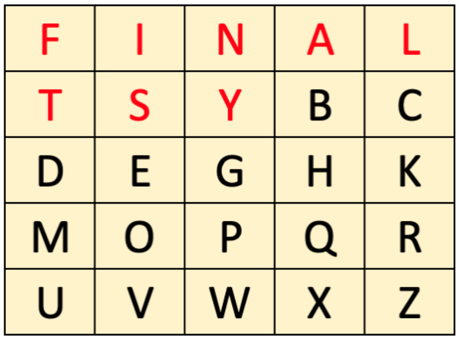
Existing modification has room for improvement as the problem in elements, and its matrix persists. There is a need to design a more efficient scheme through a dynamic character substitution process to address the vulnerability of existing Playfair to cryptanalysis due to ciphertext patterns because of its static matrix, to increase avalanche effect, and to avoid the use of non-printable characters that will ultimately produce more optimal security. This study investigates the performance and acceptability level of the proposed algorithm extent on strict avalanche criterion standard requirements.

II. LITERATURE REVIEW

A. Playfair Algorithm

The Playfair algorithm is a symmetric digraph-based substitution cipher [21]. It uses a 5x5 substitution matrix constructed using a key and the Latin alphabet characters in rows and columns [40]. Due to space constraints, the letter J is typically discarded from the grid, and the letter I is substituted in place, thus, treating them as a single character [30], [41].

Traditionally, a matrix is generated using a secret key to encrypt a plaintext. Each key character is arranged in the matrix without duplicating any letters. The remaining letters of the alphabet are added to the matrix in order. For instance, the words "FINAL FANTASY" is used as the key to creating the substitution matrix, as shown in Fig. 1.



F	I	N	A	L
T	S	Y	B	C
D	E	G	H	K
M	O	P	Q	R
U	V	W	X	Z

Fig. 1 The traditional 5x5 Playfair matrix

The algorithm works using digraphs. The plaintext is divided into pairs of characters wherein each pair must not be of similar letters. Otherwise, a substitute letter, typically X, is placed after the first letter to form another digraph. For instance, "BELLS" is represented as "BE LX LS." If the last digraph of the sequence is incomplete, then the letter 'X' is padded to complete the pair. For instance, "GIL" is represented as "GI LX." Using the matrix and each digraph, the following rules [42] are followed during substitution:

- If the letters appear on the same row of the matrix, then the ciphertext equivalent of the letters are at their right position. For instance, the digraph "FN" is substituted as "IA."
- If the letters lie in the same column of the matrix, then the ciphertext equivalent of the letters are those that are immediately below them. For instance, the digraph "TM" is substituted as "DU."
- If the letters are not found in the same row or column, then the ciphertext equivalent of the letters is those on the opposite corners of a rectangle formed among the letters. For instance, the digraph "SQ" is substituted as "BO."

It is also essential to note that the first letter of the encrypted digraph should be the one that lies on the same row as the first letter of the plaintext digraph [43]. The final ciphertext is generated once all digraphs are encoded.

To perform decryption, the same process is executed. First, the substitution matrix should be created using the key as it is difficult to decrypt the ciphertext without it. Take each ciphertext digraph and perform the substitution based on the abovementioned rules. The original plaintext is uncovered after all the digraphs are decoded.

B. Drawbacks of the Existing Playfair Algorithm

The traditional Playfair algorithm does not support numeric characters and special symbols encoding. Further, it only allows a single letter case: either upper or lower. These problems are rooted in the limited 5x5 substitution matrix, thus, restricting the keyspace to only include twenty-five (25) characters from the Latin alphabet while discarding the remaining one (1) character [44]–[46].

The Playfair algorithm is susceptible to cryptanalysis attacks such as frequency analysis and brute force [47]. The problem is traceable to the static matrix used during the substitution process. In turn, it manifests patterns in the generated ciphertext during the encryption process. Plaintext containing similar characters like 'AAAAAAAA' or 'ABABABAB,' or those that have the reverse of digraphs like 'REapER' or 'DEloadED' [48], [49] would produce repetitive character patterns in its ciphertext. Ciphertext containing these obvious patterns is easily decrypted as soon as a single part of it is already uncovered [50].

Including non-printable characters in substitution matrices also poses a problem to the encoding and decoding process. Non-printable characters are difficult to display on screens and may negatively affect the ciphertext structure [31], [32]. Using non-printable characters often leads to corruption or data loss [33], [34]. Furthermore, the transfer of non-printable characters is impossible in some communication lines [35]. Ciphertext with non-printable characters, such as control sequences, cannot be keyed in, hence, making decryption impossible for end-users [36].

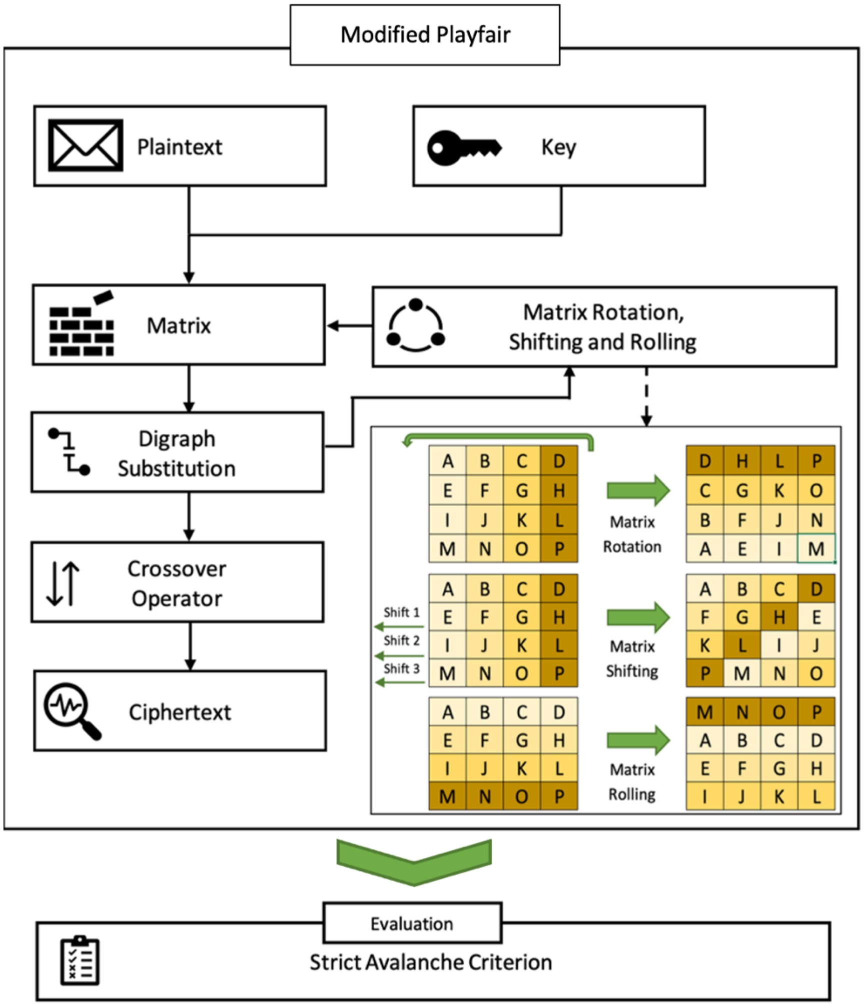
III. METHODOLOGY

A. The Proposed Enhanced Playfair Algorithm

The enhanced Playfair algorithm comprises different steps in character sequencing before substitution, as presented in Fig. 2.

The study enhances the substitution process of the traditional Playfair algorithm by introducing the Multidimensional Element-in-Grid Sequencer (MEGS) with specific operations, namely matrix rotation, shift rows, and roll to produce a dynamic substitution matrix. The use of a dynamic substitution matrix allows a more diverse character substitution keyspace against a static matrix that uses the same characters for substituting similar plaintext letters. With the implementation of a dynamic matrix, a character may be represented in 255 different ways. Moreover, a crossover operation is added for a more diverse ciphertext.

The improvement is generally achieved through the 16x16 grid and sets of operations to rearrange the grid elements. The proposed enhancement on the confusion and diffusion properties of the MEGS-based Playfair algorithm is expected to achieve a sufficient avalanche effect according to the Strict Avalanche Criterion (SAC).



A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

D	H	L	P
C	G	K	O
B	F	J	N
A	E	I	M

A	B	C	D
F	G	H	E
K	L	I	J
P	M	N	O

M	N	O	P
A	B	C	D
E	F	G	H
I	J	K	L

Fig. 2 Conceptual framework

One of the proposed enhancements features a unique matrix where three operations are introduced to arrange the 256 characters in the grid dynamically. These operations include matrix rotation, shift rows, and roll. The graphical representation of these operations is shown in Fig. 3.

The matrix rotation operation involves rotating the grid elements to a certain degree. Fig. 3 shows that the rotation follows a ninety degrees (90°) counterclockwise movement. This method transforms rows to columns and columns to rows. It can be observed that the last column containing the elements "D-H-L-P" has been transposed and has then formed the first row of the grid.

On the other hand, matrix shifting performs a cyclical shift of elements in each grid row based on a certain offset value. The shift row process starts at the second row of the grid by moving the row elements to the left one (1) time.

Succeeding rows also shift several times based on their row number minus one. They give an example showing that the values under columns A, B, C, and D are completely different from the original grid.

The third operation, called matrix roll, cyclically shifts an array of elements based on a given axis. In the proposed method, the roll operation moves an entire row downwards. As observed, the last row containing the elements "M-N-O-P" is rolled downwards and has then occupied the first row of the grid. These operations are critical in improving the confusion and diffusion properties of the Playfair algorithm. In terms of confusion, the substitution process of the proposed method completely obscures plaintext patterns in the ciphertext. The diffusion has also improved as any small changes in the plaintext result in more than half of the bits of the generated ciphertext. The confusion and diffusion properties increase are attributed to implementing a dynamic matrix.

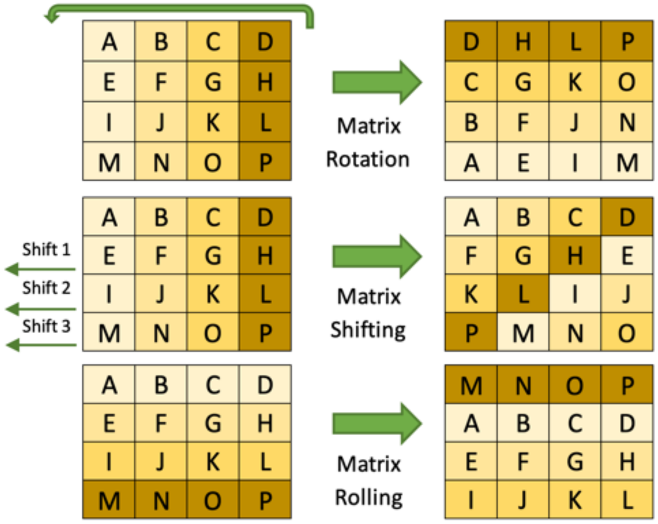


Fig. 3 Matrix operations

To illustrate how the proposed method works, a graphical representation of the encryption process is presented in Fig. 4.

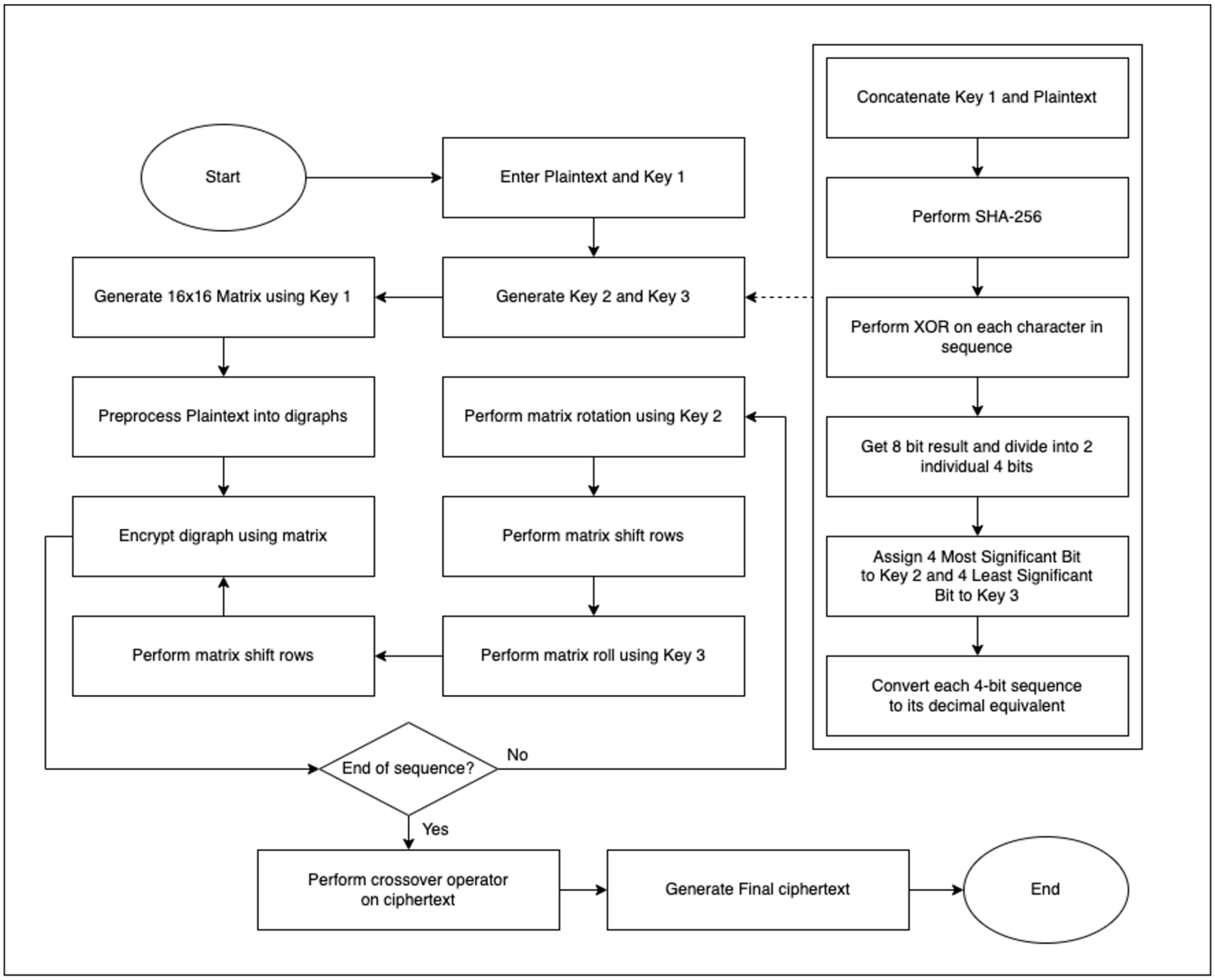

Start
Enter Plaintext and Key 1
Generate Key 2 and Key 3
Generate 16x16 Matrix using Key 1
Preprocess Plaintext into digraphs
Encrypt digraph using matrix
Perform matrix shift rows
Perform matrix rotation using Key 2
Perform matrix shift rows
Perform matrix roll using Key 3
End of sequence?
No
Yes
Perform crossover operator on ciphertext
Generate Final ciphertext
End
Concatenate Key 1 and Plaintext
Perform SHA-256
Perform XOR on each character in sequence
Get 8 bit result and divide into 2 individual 4 bits
Assign 4 Most Significant Bit to Key 2 and 4 Least Significant Bit to Key 3
Convert each 4-bit sequence to its decimal equivalent

Fig. 4 Proposed encryption process

First, plaintext and key 1 are required as input. The plaintext is the message to be encrypted, while key 1 is used to generate the initial Playfair substitution matrix. Before the matrix operations are performed, key 2 and key 3 must be obtained. The two additional keys are used to specify the number of times a matrix operation should be executed. To identify key 2 and key 3 values, the plaintext and key 1 are concatenated and then hashed using SHA-256. An exclusive-OR (XOR) operation is performed on each succeeding character of the hash until the end of its sequence. The resulting 8-bit sequence is split into two 4-bit sequences. The four (4) most significant bits are assigned to key 2, and the other four (4) least significant bits to key 3. Both keys are converted into their decimal equivalent.

Encryption is executed after setting the values of key 2 and key 3. Using key 1, an initial substitution matrix is generated. The matrix is populated using key 1 to appear in the key and without placing any duplicate characters. Subsequently, the remaining printable ASCII characters which are not part of the key will be added to the matrix. To start encrypting, the plaintext is divided into several digraphs. Each of these digraphs is encrypted using a dynamic substitution matrix. The matrix elements are rearranged for each digraph encrypted to form a grid of elements. Several matrix operations are performed in rotate, shift rows, roll, and shift rows. The number of matrix rotations is based on the value of the key, while the number of matrix rolls is based on the value of key 3. Once the elements have been rearranged, the next digraph is taken and then encrypted using the new matrix. The same process is repeatedly executed until the last digraph has been encoded.

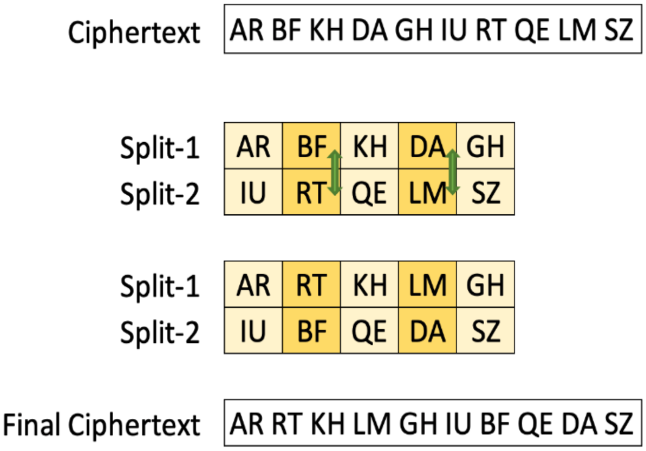


Fig. 5 Crossover operation

Finally, the ciphertext is produced by performing a multipoint crossover operation using the encrypted digraphs. The sequence of digraphs is split in half, and all digraphs indexed in even positions in the sequence are swapped. The sample crossover operation is shown in Fig. 5.

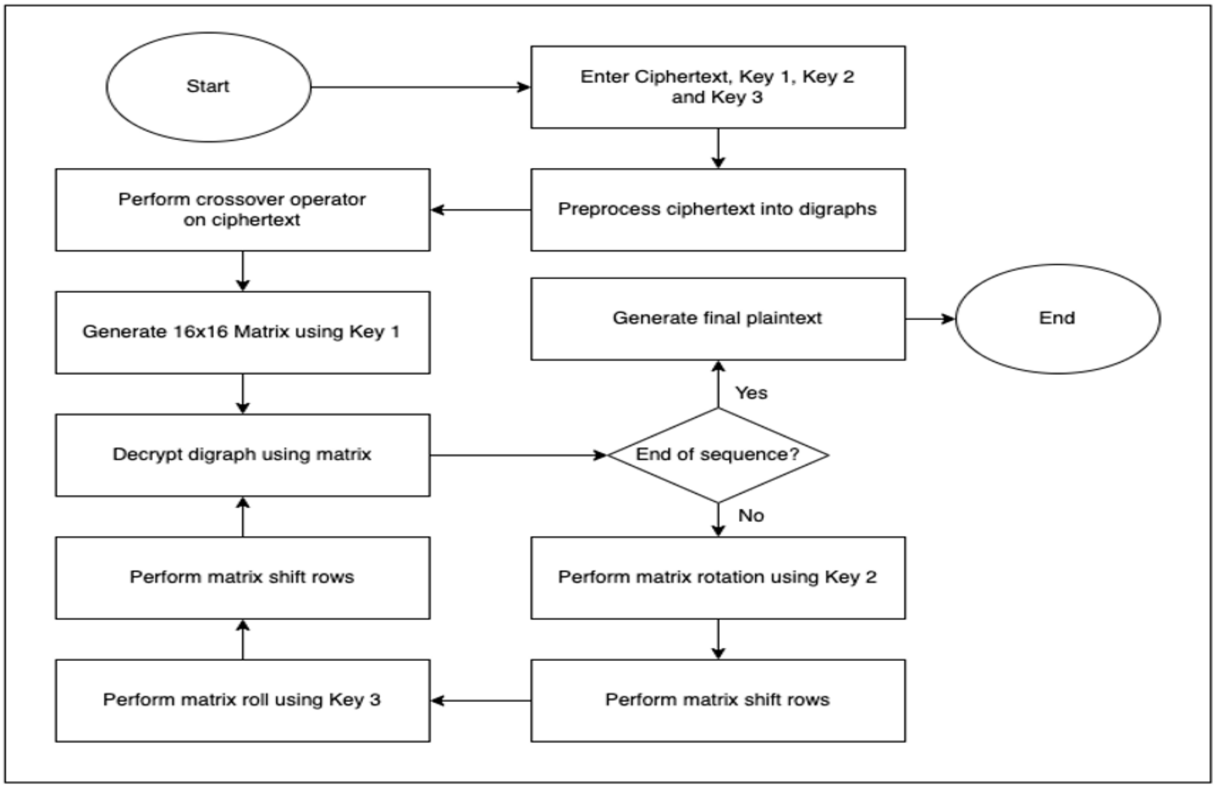

Start
Enter Ciphertext, Key 1, Key 2 and Key 3
Perform crossover operator on ciphertext
Preprocess ciphertext into digraphs
Generate 16x16 Matrix using Key 1
Generate final plaintext
End
Yes
Decrypt digraph using matrix
End of sequence?
No
Perform matrix shift rows
Perform matrix rotation using Key 2
Perform matrix roll using Key 3
Perform matrix shift rows

Fig. 6 Proposed decryption process

To execute the decryption process, the ciphertext with keys 1, 2, and 3 are required as input. First, the ciphertext undergoes a multipoint crossover operation. The ciphertext is split into digraphs, and then the sequence of digraphs is split in half. All digraphs that fall in even positions are swapped. Next, key 1 is used to generate the initial 16x16 matrix. Digraphs are decrypted using the matrix through character substitution. For each digraph decrypted, the matrix elements are rearranged to form a new grid of elements through matrix operations performed in the order of rotate, shift rows, roll, and shift rows. The number of matrix rotations and matrix rolls is based on key 2 and key 3, respectively. Once the elements have been rearranged, the next digraph is taken and then decrypted using the new matrix. The same process is repeatedly executed until the last digraph has been decoded. The detailed step of the decryption process is shown in Fig. 6.

B. Method Evaluation

To test the security of the proposed method, the avalanche effect was considered the evaluation technique. The proposed algorithm must meet the Strict Avalanche Criterion (SAC) standard.

The avalanche effect is an essential factor in determining the success of a cryptographic algorithm. The avalanche effect is calculated by computing the confusion element after altering just one bit from the plaintext that significantly changes the ciphertext. The desired value of the avalanche effect is 0.5 [43]. The avalanche effect is calculated using (1):

$$AE(\%) = \frac{\#\ of\ Changed\ Bits\ in\ Ciphertext}{Total\ \#\ of\ Bits\ in\ Ciphertext} x100 \qquad (1)$$

IV. RESULTS AND DISCUSSION

The simulation of the MEGS-based Playfair algorithm was executed in macOS version 12 environment using Python 3.8 and PyCharm IDE Community Edition 2021 in a MacBook Pro with a 2.0 GHz Quad-Core i5 processor, 16GB RAM, and Intel Iris Plus Graphics video card.

The avalanche effect of the MEGS-based Playfair algorithm is shown in Table 1.

Table 1. Average avalanche effect using plaintext with varied lengths

Sample	Length of Plaintext	Avalanche Effect
1	10 characters	55.47%
2	20 characters	58.44%
3	30 characters	55.30%
4	40 characters	57.42%
5	60 characters	55.72%
6	75 characters	52.96%
7	100 characters	52.76%
8	150 characters	52.07%
9	350 characters	52.24%
10	450 characters	52.70%
11	512 characters	51.88%
12	600 characters	51.48%
13	800 characters	51.50%
14	900 characters	51.65%
15	1,000 characters	51.52%
Average		**53.54%**

Plaintext with varying lengths ranging from 10 to 1000 was simulated to obtain each avalanche effect. Every character of the given plaintexts was flipped by one bit iteratively to determine the results. The average avalanche effect of the fifteen samples using plaintext with different lengths is 53.54%, satisfying the minimum percentage requirement of at least 50% for a desirable avalanche effect of a cryptographic algorithm. This denotes that the MEGS-based Playfair algorithm produces a difficult-ciphertext to uncover using statistical analyses [51]–[53].

V. CONCLUSION AND RECOMMENDATION

In this study, the use of a 16x16 dynamic matrix with the Multidimensional Element-in-Grid Sequencer (MEGS) for the Playfair algorithm was implemented. The matrix rotation, matrix shifting, matrix rolling, and crossover operations were introduced in the construction of the dynamic matrix, which paved the way for a more secure encryption process. The simulation results revealed that the proposed MEGS-based Playfair algorithm yielded an average of 53.54% avalanche effect for all given plaintexts with varying lengths. The results denote that the algorithm successfully passed the strict avalanche criterion standard requirement for a desirable cryptographic algorithm. It is recommended to utilize the proposed algorithm in password security or image steganography and the like for future works. It is also suggested that the modified Playfair algorithm be tested using other performance metrics in cryptography, such as the randomness test and brute force attack analysis.

REFERENCES

[1] W. Stallings, Cryptography and Network Security Principles and Practices. Prentice-Hall, (2005).

[2] M. H. Saracevic et al., Data Encryption for Internet of Things Applications Based on Catalan Objects and Two Combinatorial Structures, IEEE Trans. Reliab., 70(2) (2021) 819–830 doi: 10.1109/TR.2020.3010973.

[3] A. Rajab, S. Aqeel, M. S. Al Reshan, A. Ashraf, S. Almakdi, and K. Rajab, Cryptography-based encryption techniques for the security of data in cloud computing paradigm, Int. J. Eng. Trends Technol., 69(10) (2021) 1–6. doi: 10.14445/22315381/IJETT-V69I10P201.

[4] J. V. Karthik and B. V. Reddy, Authentication of secret information in image steganography, Int. J. Latest Trends Eng. Technol., 3(1)(2013) 97–104. doi: 10.1109/TENCON.2008.4766581.

[5] R. Deepthi, A Survey Paper on Playfair Cipher and its Variants, Int. Res. J. Eng. Technol. 4(4)(2017) 2607–2610.

[6] M. Syahrizal, M. Murdani, S. D. Nasution, M. Mesran, R. Rahim, and A. P. U. Siahaan, Modified Playfair Cipher Using Random Key Linear Congruent Method,(2017).

[7] R. Rahim and A. Ikhwan, Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher, Int. J. Sci. Res. Sci. Technol., 2(6) (2016) 71–78.

[8] H. B. Macit, A. Koyun, and M. E. Yüksel, Embedding Data Crypted With Extended Shifting Polybius Square Supporting Turkish Character Set, BEU J. Sci., 8(1)(2019) 234–242.

[9] E. V. Haryannto, M. Zulfadly, Daifiria, M. B. Akbar, and I. Lazuly, Implementation of Nihilist Cipher Algorithm in Securing Text Data With Implementation of Nihilist Cipher Algorithm in Securing Text Data With Md5 Verification, J. Phys. Conf. Ser.1361, (01) (2020) (2019), doi: 10.1088/1742-6596/1361/1/012020.

[10] G. Manikandan, P. Rajendiran, R. Balakrishnan, and S. Thangaselvan, A Modified Polybius Square Based Approach for Enhancing Data Security, Int. J. Pure Appl. Math.119(12) (2018) 13317–13324.

[11] M. Maity, A Modified Version of Polybius Cipher Using Magic Square and Western Music Notes, Int. J. Technol. Res. Eng., 1,(10) (2014) 1117–1119.

[12] C. Kumar, S. Dutta, and S. Chakraborty, A Hybrid Polybius-Playfair Music Cipher A Hybrid Polybius-Playfair Music Cipher, Int. J. Multimed. Ubiquitous Eng.,10(8),(2015)187–198. doi: 10.14257/ijmue.2015.10.8.19.

[13] A. Singh and S. Sharma, Enhancing Data Security in Cloud Using Split Algorithm, Caesar Cipher, and Vigenere Cipher, Homomorphism Encryption Scheme, in Emerging Trends in Expert Applications and Security (841) (2019) 157–166. doi: 10.1007/978-981-13-2285-3.

[14] I. Gunawan, Sumarno, H. S. Tambunan, E. Irawan, H. Qurniawan, and D. Hartama, Combination of Caesar Cipher Algorithm and Rivest Shamir Adleman Algorithm for Securing Document Files and Text Messages, J. Phys. Conf. Ser., 1255, (2019). doi: 10.1088/1742-6596/1255/1/012077.

[15] D. Gautam, C. Agrawal, P. Sharma, M. Mehta, and P. Saini, An Enhanced Cipher Technique Using Vigenere and Modified Caesar Cipher,(2018), doi: 10.1109/ICOEI.2018.8553910.

[16] A. Banerjee, M. Hasan, and H. Kafle, Secure Cryptosystem Using Randomized Rail Fence Cipher for Mobile Devices, in Intelligent Computing - Proceedings of the Computing Conference, (2019)737–750, doi: 10.1007/978-3-030-22868-2.

[17] A. P. U. Siahaan, Rail Fence Cryptography in Securing Information, Int. J. Sci. Eng. Res., 7(7)(2016) 535–538.

[18] A. C. Licayan, B. D. Gerardo, and A. A. Hernandez, Enhancing Playfair Cipher using Seed Based Color Substitution, Proc. - 2020 16th IEEE Int. Colloq. Signal Process. Its Appl. CSPA 2020,(2020)242–246. doi: 10.1109/CSPA48992.2020.9068730.

[19] R. Patil, S V Bang, and R. B. Bangar, Improved Cryptography by Applying Transposition on Modified Playfair Algorithm Followed by Steganography, Int. J. Innov. Sci. Res. Technol., 6(5) (2021) 616–620.

[20] M. I. Jabiullah, A. D. Arni, and B. B. Brishti, A Playfair Cipher-based Secured Patients ' Information Transaction System, J. Netw. Secur. Data Min., 4(1)(2021)1–6. doi: 10.5281/zenodo.4699968.

[21] N. Chand, S. Bhattacharyya, and A. Sarkar, A Novel Encryption Technique to Protect Patient Health Information Electronically Using Playfair Cipher 15 by 14 Matrix, in Advances in Medical Physics and Healthcare Engineering, (2021)423–431.

[22] K. J. Devi, P. Singh, R. K. Yadav, and M. Z. Gafaru, Reversible and Secured Image Watermarking Technique for IoMT Healthcare, in 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), (2021)1–6, doi: 10.1109/ICCCNT51525.2021.9580071.

[23] M. R. Sani, A Play-Fair Cipher Based Authentication Medical Data Image Transaction Process,(2019).

[24] R. Saharan and S. Yadav, A Novel Method for Image Encryption, Inf. Commun. Technol. Intell. Syst. Smart Innov. Syst. Technol., 106(2019)427–434.doi: 10.1007/978-981-13- 1742-2_42.

[25] H. R. Shakir and S. A. Yassir, Image Encryption-Compression Method Based on Playfair, OTP and DWT for Secure Image Transmission, in Advances in Cyber Security. ACeS 2021. Communications in Computer and Information Science, , 1487(2021) 95–113. doi: 10.1007/978-981-16-8059-5_7.

[26] M. Saraswathi and K. N. Meera, An Application of Radio Mean Labeling in Cryptography, (2021), doi: 10.1109/phdedits53295.2021.9649460.

[27] J. David Livingston and E. Kirubakaran, Implementation of extended play-fair algorithm for client-side encryption of cloud data, Intell. Big Data Technol. Hype. Adv. Intell. Syst. Comput., 1167(2021). 485–493. doi: 10.1007/978-981-15-5285-4_48.

[28] S. Karthiga and T. Velmurugan, Enhancing Security in Cloud Computing using Playfair and Ceasar Cipher in Substitution Techniques, Int. J. Innov. Technol. Explore. Eng., 9(4)(2020) 912–920. doi: 10.35940/ijitee.d1363.029420.

[29] B. Priyatna and A. L. Hananto, Password Data Authentication Using a Combination of MD5 and Playfair Cipher Matrix 13x13, Buana Inf. Technol. Comput. Sci. (BIT CS),1(2)(2020) 33–36. doi: 10.36805/bit-cs.v1i2.980.

[30] R. K. Salih and M. S. Yousif, Playfair with Multi Strata Encryption, Iraqi J. Sci.,62(9)(2021)3237–3242. doi: 10.24996/ijs.2021.62.9.36.

[31] E. Zadok, I. Badulescu, and A. Shender, Cryptfs: A Stackable Vnode Level Encryption File System, Columbia Univ. Acad. Commons, (1998) doi: https://doi.org/10.7916/D82N5935.

[32] P. Guldin and Y. Zhuge, What Do You Mean My CSV Doesn't Match My SAS® Dataset?, in SAS Conference Proceedings: SouthEast SAS Users Group 2016, 2016, pp. 1–10, [Online]. Available: https://analytics.ncsu.edu/sesug/2016/CC-132_Final_PDF.pdf.

[33] K. Berisso, Reed-Solomon Based Bar Code Character Substitution Rates,J. Bus. Manag. Sci., v 6(3)(2018). 70–75. doi: 10.12691/jbms-6-3-1.

[34] D. Moses and C. Deisy, A Novel Lossless ECG Compression Technique for Transmission in GSM Networks, in Proceedings of the Third International Conference on Soft Computing for Problem Solving - Advances in Intelligent Systems and Computing, 259(2014), doi: 10.1007/978-81-322-1768-8.

[35] S. J. Sarkar, P. K. Kundu, and G. Sarkar, Realization of a Cloud Server Based Power System Operational Data Management System, in ICECE 2018 - 10th International Conference on Electrical and Computer Engineering, (2019)105–108, doi: 10.1109/ICECE.2018.8636816.

[36] Z. A. Genç, G. Lenzini, and P. Y. A. Ryan, Security analysis of key acquiring strategies used by cryptographic ransomware, in CECC 2018: Proceedings of the Central European Cybersecurity Conference 2018 - ACM International Conference Proceeding Series, (2018)1–6, doi: 10.1145/3277570.3277577.

[37] A. Singh, P. Agarwal, and M. Chand, Image Encryption and Analysis using Dynamic AES, 2019 Int. Conf. Optim. Appl. ICOA 2019, (2019) 1–6. doi: 10.1109/ICOA.2019.8727711.

[38] S. S. Hameed, SMX Algorithm : A Novel Approach to Avalanche Effect on Advanced Encryption Standard AES, 5th Int. Conf. Computing Sustain. Glob. Dev.(2018)727–732.

[39] I. A. Shoukat, U. Iqbal, A. Rauf, and M. R. Faheem, Randomized Substitution Method for Effectively Secure Block Ciphers in I.O.T Environment,Arab. J. Sci. Eng.45(12)(2020) 11019–11036. doi: 10.1007/s13369-020-04919-3.

[40] N. Jain and S. S. Chauhan, Novel Approach Transforming Stream Cipher to Block Cipher, in 2021 International Conference on Technological Advancements and Innovations (ICTAI), (2021) 182–187, doi: 10.1109/ICTAI53825.2021.9673175.

[41] C. Sharma, A. Kumar, A. Sinha, and M. Ahmad, A Novel Encryption RAAM Algorithm in Different Multimedia Applications, Int. J. Soft Comput. Eng., 10(5) (2021) 9–13. doi: 10.35940/ijsce.f3492.0510521.

[42] M. M. Maha, M. Masuduzzaman, and A. Bhowmik, An effective modification of play fair cipher with performance analysis using 6X6 matrix, in ICCA 2020: Proceedings of the International Conference on Computing Advancements, (2020) 1–6, doi: 10.1145/3377049.3377085.

[43] R. M. Marzan, A. M. Sison, and R. P. Medina, An Enhanced Key Security of Playfair Cipher Algorithm,(2019), doi: 10.1145/3316615.3316689.

[44] G. Sharma, P. Goyal, and S. S. Kushwah, Implementation of Modified Playfair CBC Algorithm, Int. J. Eng. Res., 5(06) (2016) 679–684. doi: 10.17577/ijertv5is060631.

[45] S. A. Noaman, Adaptive Playfair cipher Crypto algorithm, J. Al-Qadisiyah Comput. Sci. Math., 9(2) (2017) 114–121. doi: 10.29304/jqcm.2017.9.2.320.

[46] R. Deepthi, A Survey Paper on Playfair Cipher and its Variants, Int. Res. J. Eng. Technol., 4(4) (2017) 2607–2610.[Online].Available: https://www.irjet.net/archives/V4/i4/IRJET-V4I4642.pdf.

[47] A. Sharma, N. Gupta, A. Thakur, K. Guleri, and M. Dhiman, Enhancing Communication Using 8 × 8 Extended Playfair Cipher and Steganography,(2020). doi: 10.36227/techrxiv.12287483.v1.

[48] S. M. Hardi, J. T. Tarigan, and N. Safrina, Hybrid cryptosystem for an image file using elgamal and double Playfair cipher algorithm, in Journal of Physics: Conference Series,978 (2018). doi: 10.1088/1742-6596/978/1/012068.

[49] S. S. Srivastava and N. Gupta, A Novel Approach to Security using Extended Playfair Cipher, Int. J. Comput. Appl.20(6)(2011)39–43. doi: 10.5120/2435-3276.

[50] J. C. T. Arroyo and A. J. P. Delima, An Improved Affine Cipher using Blum Blum Shub Algorithm, Int. J. Adv. Trends Comput. Sci. Eng. 9(3)(2020) 3295–3298. doi: 10.30534/ijatcse/2020/126932020.

[51] J. Lee, N. Sultana, F. Yi, and I. Moon, Avalanche and bit independence properties of photon-counting double random phase encoding in gyrator domain, Curr. Opt. Photonics,2(4)(2018) 368–377. doi: 10.3807/COPP.2018.2.4.368.

[52] N. Kapalova and A. Haumen, The model of encryption algorithm based on non-positional polynomial notations and constructed on an SP-network, Open Eng.,8(1) (2018) 140–146. doi: 10.1515/eng-2018-0013.

[53] X. Tong, M. Cui, and Z. Wang, A new feedback image encryption scheme based on perturbation with dynamical compound chaotic sequence cipher generator, Opt. Commun.,282(14) (2009) 2722–2728. doi: 10.1016/j.optcom.2009.03.075.