A Novel ASCII Code-based Polybius Square Alphabet Sequencer as Enhanced Cryptographic Cipher for Cyber Security Protection (APSAlpS-3CS)

Jan Carlo T. Arroyo[1], Ariel Roy L. Reyes[2], Allemar Jhone P. Delima[3]
College of Computing Education, University of Mindanao, Davao City, Davao del Sur, Philippines[1]
College of Information and Computing, University of Southeastern Philippines, Davao City, Davao del Sur, Philippines[1, 2]
College of Engineering, Technology and Management, Cebu Technological University-Barili Campus, Cebu, Philippines[3]

***Abstract*—For all industries, cybersecurity is regarded as one of the major areas of concern that needs to be addressed. Data and information in all forms should be safeguarded to avoid leakage of information, data theft, and robbery through intruders and hackers. This paper proposes a modification on the traditional 5x5 Polybius square in cryptography, through dynamically generated matrices. The modification is done through shifting cell elements for every encrypted character using a secret key and its ASCII decimal code equivalents. The results of the study revealed that the modified Polybius cipher offers a more secure plaintext-ciphertext conversion and is difficult to break, as evident in the frequency analysis. In the proposed method, each element produced in the digraphs exhibits a wider range of possible values. However, with the increase of process in the encryption and decryption, the modified Polybius cipher obtained a longer execution time of 0.0031ms, being identified as its tradeoff. The unmodified Polybius cipher, however, obtained an execution time of 0.0005ms. Future researchers may address the execution time tradeoff of the modified Polybius cipher.**

Keywords*—*Cryptography; ciphers; ciphertext; modified polybius cipher; plaintext

I. Introduction

Cybersecurity has been the primary focus of all industries when it comes to communications security, as heterogeneous data needs to be safeguarded. As all industries leverage on the use of the internet, most of today’s transactions are carried out online as a medium of communication since mobile devices and Wi-Fis are readily available. All forms of data ranging from public to private organizations of all sizes possess valuable information and are vulnerable to attacks. Subsequently, data security implementation is still an ongoing quest, and managing information technology (I.T) security has become a challenge due to factors such as the unavailability of experts and the cost involved. Therefore, security has been a concern of all industries as companies are on war against hackers and intruders [1].

Various researches in this area lobby the use of algorithms to address security issues and provide data and information security against hackers and intruders. Cryptography [2], being a known technique in communications security, deals with algorithms for security services to wit: confidentiality and integrity of data, authentication to a wireless communication system, and several other security protocols where information transfer is done between different users [3]–[5]. With extent to the data format such as text, the data protection is made through the use of encryption where conversion of the plaintext into an unintelligible format called ciphertext is done. To revert the ciphertext, a decryption process is done. A cipher algorithm is responsible for the encryption and decryption of the data [2]. Some of the classical ciphers found in the literature are ADFGX Cipher [6]–[8], Affine Cipher [8]–[11], Atbash Cipher [12], Auto-key Cipher [13], Baconian Cipher [8], [14], Base64 Cipher [15]–[17], Beaufort Cipher [8], Caesar Cipher [18]–[20], Enigma Machine Cipher [8], Four-square cipher [8], Grille Cipher [21], Hill cipher [22], Homophonic Substitution Cipher [23], [24], Permutation Cipher [8], Playfair Cipher [25]–[27], Polybius Cipher [28]–[32], and Rail fence Cipher [33], [34], among others.

Among these, the Polybius cipher, also known as the Polybius square, is one of the commonly used methods for cryptography [35], [36]. It is one of the early cryptographic systems developed for obscuring plaintext by fractionating and substituting numbers [37]. To date, the Polybius square is still extremely valuable for cryptographers. Its ability to convert letter sequences to numeric sequences, reduce the number of different characters, and allow encoding of plaintext into two separately manipulatable units are known to be its advantages [6], [38]. Further, the Polybius square has paved the way to the development and processes of other classical ciphers that are still used today such as ADFGVX Cipher [8], [39]–[41], Bifid cipher [8], [42], Nihilist cipher [43], and Trifid cipher [44], [35]. Furthermore, modern cryptographic systems have embedded the Polybius square as a fundamental component of the cryptographic process, such as in the key generation procedures used by modern ciphers like the advance encryption standard (AES), data encryption standard (DES), and other algorithms [2], [39], [40], [45]–[50].

However, the Polybius square is a substitution cipher that is susceptible to attacks and is easy to crack with frequency analysis due to the simplicity of element distribution within its grid [51]. This problem is rooted in the structure and elements within the square grid. Therefore, there is a need to introduce a new scheme for character sequencing before performing the substitution; thus, this study. The rest of the paper is structured as follows: Section II presents the literature review of existing methodology and modifications in the Polybius cipher. The proposed enhancement in the Polybius cipher is discussed in Section III, while Section IV presents the results and discussion. The conclusion is shown in Section V.

II. Literature Review

Cryptography [4] is one of the widely used obscuring techniques to protect data and is commonly used in various industries [52]–[55]. Communications security is ensured through the use of ciphers whose bottleneck for an optimal implementation relies on the cipher algorithm used for encryption and decryption process. The basic ciphers are categorized into two: the substitution and transpositions. A transposition cipher transposes or reorders elements such that elements in the first place of the plaintext may be positioned in any other place of the ciphertext. Likewise, an element in the seventh place of the plaintext may be positioned in the first place of the ciphertext [10]. Meanwhile, substitution ciphers is an encoding technique where characters in the plaintext are replaced with a character or symbol or both. Ciphers such as ADFG(V)X, Alberti cipher, Autokey cipher, Caesar cipher, Four-square cipher, Polybius cipher, Enigma cipher, Freemason cipher, Kamasutra cipher, Larrabee cipher, Monoalphabetic substitution cipher, and Pollux cipher takes a letter of an alphabet and substitutes it with another character [51].

The Polybius cipher, along with other Polybius-based ciphers, is continuously being utilized along with other modern cryptographic ciphers to improve services that involve and require text security protection in digital media, such as for online shopping, internet banking, chip operation, mobile cloud computing, and mobile messaging services [56], [49], [50].

A. *The Traditional Polybius Cipher*

Polybius cipher, known as Polybius square, is one of the early encryption systems recorded in the history that was developed by Greek historian and a soldier, Polybius [37]. Polybius square is a substitution cipher placed in a 5x5 grid matrix where the alphabet is arranged with corresponding rows and columns without repetition. As one of the earliest ciphers developed in history, its usage was early recorded as a medium of communication and used even in wars [35].

In the Polybius square, letters in the modern English alphabet comprising of 26 characters are placed in the 5x5 grid. Individual letters are spread all throughout the 25 cells in the matrix wherein characters J and I are combined as they share the same code [57]. The Polybius square with the English alphabet is shown in Table I.

Encryption and decryption using this technique are relatively easy because there is no need for a key. In encrypting plaintext, the characters are matched to the matrix one by one to retrieve their coordinates based on the intersection of row and columns. The set of coordinates generated represents the encrypted message. For example, encrypting the word CIPHER results to 1324352342, where character C is 13, I is 24, and so on. Table II shows the encryption result using the traditional Polybius square.

To decrypt a given ciphertext, the process is done in reverse. Each pair of numbers are compared to the matrix to translate the value to their corresponding plaintext form. In this case, the encrypted message 1324352342 is converted to CIPHER. Table III shows the encryption result using the traditional Polybius square.

Polybius cipher has been the basis of some of today's encryption methods. However, the Polybius square has its identified drawbacks. The cipher does not have a key for data encryption and decryption process making it vulnerable for cracks [34], [58]. Moreover, adding the characters I and J in one cell may cause complications to the original plaintext; hence, it may confuse the decoding process [47]. The ciphertext using the Polybius square is easy to decipher as characters are always represented as a pair.

The following sections discuss the modifications and hybridization made on the Polybius Cipher.

B. *A Modified Polybius Square Based Approach for Enhancing Data Security*

The study of [30] introduced a 6x6 grid Polybius square to include the English alphabet and numbers. In this square matrix, the numbers are first encoded, followed by the English alphabet, as shown in Table IV.

TABLE I. POLYBIUS SQUARE WITH THE MODERN ENGLISH ALPHABET

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

TABLE II. ENCRYPTION USING TRADITIONAL POLYBIUS SQUARE

Plaintext	C	I	P	H	E	R
Position	1	2	3	4	5	6
Ciphertext	13	24	35	23	15	42

TABLE III. DECRYPTION USING TRADITIONAL POLYBIUS SQUARE

Ciphertext	13	24	35	23	15	42
Position	1	2	3	4	5	6
Plaintext	C	I	P	H	E	R

TABLE IV. POLYBIUS SQUARE WITH DIGITS

0	1	2	3	4	5
6	7	8	9	a	b
c	d	e	f	g	h
i	j	k	l	m	n
o	p	q	r	s	t
u	v	w	x	y	z

Techniques such as transposition, ring rotation, and row reversal are introduced to transmute the matrix. First, the grid performs row reversal by swapping the values in the row such that the 1st element becomes the 5th element, and is reciprocated. The same is applied for each remaining element in the row. For example, row 1 with elements 0|1|2|3|4|5 becomes 5|4|3|2|1 Table V shows the resulting elements' arrangement after applying row reversal.

The next process is the transposition, where values in each corresponding row are rewritten in columns. For example, row 1 with elements 5|4|3|2|1 is rewritten in column 1 with the same values. The new elements arrangement within the square grid after transposition is shown in Table VI.

Lastly, the matrix performs a rotation based on a given key. For example, with the given key as SASTRA, the key is used to retrieve the number of rotations by finding the sum of its ASCII values modulo length of the ring. Since the ASCII sum for the key is 654, the outermost ring is rotated by 14 times clockwise (654%20=14), while the second outermost ring and the inner ring is rotated 6 (654%12=6) and 2 (654%4=2) times, respectively. The grid matrix with new elements after the ring rotation is shown in Table VII.

After completing the processes, every character must be identified according to their relative coordinate in the new matrix and then crosschecked with the equivalent value in the original matrix using its coordinates. For instance, the element A in the new matrix is at coordinates (5,5); therefore, the plaintext is replaced with ciphertext S based on the given value in the same position from the original matrix. Table VIII shows the encryption result of the given key SASTRA.

C. An Extended Version of the Polybius Cipher

In the quest to include symbols and numbers in the Polybius square, the 5x5 grid from the traditional Polybius square matrix was expanded into an 8x8 grid matrix. With this expansion, a wide range of characters, symbols, and numbers have been used in encrypting messages. A keyword was also introduced to adjust the character's arrangement in the matrix. The keyword is placed on the top cells reaching the bottom and left to right cells of the grid without repetitions. Any remaining letters that are not used in the keyword are placed in the remaining cells in alphabetical order. Further, numbers are positioned in ascending order, followed by the special symbols arranged according to their ASCII value. Table IX shows how the extended Polybius square would appear using the keyword POLY2013 [47].

D. A Hybrid Polybius-Playfair Music Cipher

The paper [32] introduced a hybrid Polybius and Playfair cipher that translates plaintext into musical notes. To execute, the message is converted using Playfair digraphs and is encrypted with a key. The Polybius square with Playfair key labeled with five major music chords ABCDE is shown in Table X.

The generated ciphertext using the Playfair is re-encrypted using the Polybius square. The ciphertext is the musical equivalents of the Polybius cipher. Table XI shows how the string HELLO WORLD is encrypted using the hybrid technique. Performing the process in reverse order converts the ciphertext to the original plaintext by matching the chords with the generated matrix.

TABLE V. POLYBIUS SQUARE AFTER ROW REVERSAL PROCESS

5	4	3	2	1	0
b	a	9	8	7	6
h	g	f	e	d	c
n	m	l	k	j	i
t	s	r	q	p	o
z	t	x	w	v	u

TABLE VI. POLYBIUS SQUARE AFTER TRANSPOSITION PROCESS

5	b	h	n	t	z
4	a	g	m	s	y
3	9	f	l	r	x
2	8	e	k	q	w
1	7	d	j	p	v
0	6	c	i	o	u

TABLE VII. POLYBIUS SQUARE AFTER RING ROTATION PROCESS

Y	x	w	v	U	o
Z	p	j	d	7	i
T	q	k	e	8	c
N	r	l	f	9	6
H	s	m	g	A	0
B	5	4	3	2	1

TABLE VIII. ENCRYPTION USING THE MODIFIED POLYBIUS SQUARE

Plaintext	s	a	s	t	r	a
Coordinates	(5,2)	(5,5)	(5,2)	(3,1)	(4,2)	(5,5)
Ciphertext	p	s	p	c	j	s

TABLE IX. EXTENDED POLYBIUS SQUARE MATRIX

	1	2	3	4	5	6	7	8
1	P	O	L	Y	2	0	1	3
2	A	B	C	D	E	F	G	H
3	I	J	K	M	N	Q	R	S
4	T	U	V	W	X	Z	4	5
5	6	7	8	9		!	“	#
6	$	%	&	‘	(	)	*	+
7	,	-	.	/	:	;	<	=
8	>	?	@	[	\	]	^	_

TABLE X. HYBRID POLYBIUS-PLAYFAIR KEY GRID MATRIX

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

TABLE XI. ENCRYPTION USING POLYBIUS-PLAYFAIR CIPHER

Plaintext	HELLO WORLD					
Playfair Digraphs	HE	LX	LO	WO	RL	DX
Playfair Cipher	KG	YV	RV	VQ	GR	ZC
Polybius Cipher	CDCB	ADEB	BBEB	EBDC	CBBB	EEBD

III. PROPOSED ENHANCED POLYBIUS CIPHER

In this study, the sequencing of the elements within the grid is modified. Instead of using the traditional character sequence in assigning coordinates as the ciphertext, the use of ASCII code to transmute the elements in the matrix with a secret key is introduced.

The enhanced polybius square (EPS) works by altering the arrangement of elements in the matrix based on the individual characters of a given key. For every character encrypted, a new matrix is used; therefore, similar plaintext letters may not have the same encryption value. This ensures that the ciphertext produced by the modified technique is always dynamic and more complicated to crack using frequency analysis.

The steps to encrypt a message using EPS is presented in Fig. 1, where detailed steps are as follows:

a) Identify a plaintext message and a secret key.

b) Each character from the secret key is paired with each of the plaintext characters. If the length of the secret key is less than the plaintext, it is paired repeatedly in a circular manner until the end of the plaintext length.

c) Take character n from the plaintext and its corresponding key pair.

d) Convert the key character to its ASCII decimal equivalent.

e) Perform a right shift to the elements of the Polybius Square based on the ASCII decimal equivalent.

f) Generate the ciphertext equivalent of character n using the transmuted matrix.

g) Repeat steps c to f until all characters in the plaintext are converted.

The steps to decrypt a message using EPS are presented in Fig. 2, where detailed steps are as follows:

a) Identify the ciphertext message and the secret key used.

b) Each character from the secret key is paired with each of the ciphertext digraphs. If the length of the secret key is less than the ciphertext, it is paired repeatedly in a circular manner until the end of the ciphertext length.

c) Take digraph n from the ciphertext and its corresponding key pair.

d) Convert the key character to its ASCII decimal equivalent.

e) Perform a right shift to the elements of the Polybius Square based on the ASCII decimal equivalent.

f) Generate the plaintext equivalent of digraph n using the transmuted matrix.

g) Repeat steps c to f until all digraphs in the ciphertext are converted.

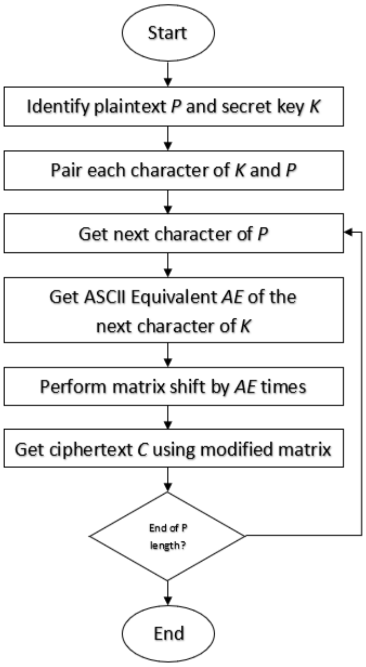
Start
Identify plaintext *P* and secret key *K*
Pair each character of *K* and *P*
Get next character of *P*
Get ASCII Equivalent *AE* of the next character of *K*
Perform matrix shift by *AE* times
Get ciphertext *C* using modified matrix
End of P length?
End

Fig. 1. Encryption Process

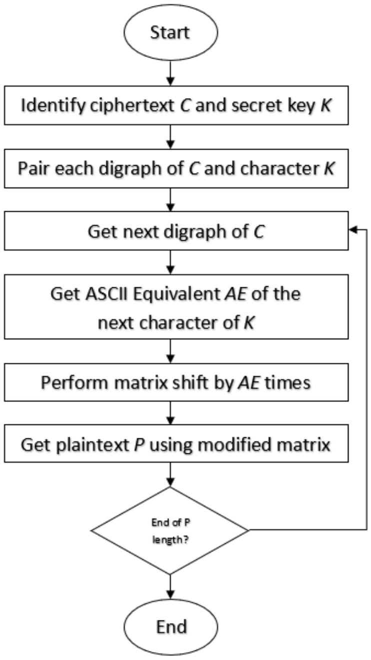
Start
Identify ciphertext *C* and secret key *K*
Pair each digraph of *C* and character *K*
Get next digraph of *C*
Get ASCII Equivalent *AE* of the next character of *K*
Perform matrix shift by *AE* times
Get plaintext *P* using modified matrix
End of P length?
End

Fig. 2. Decryption Process.

IV. Results and Discussion

A. Enhanced Polybius Square Simulation Results

In this study, a simple program was created using Python 3 and was executed in an i7-7700HQ 2.80GHz 16GB RAM 4GB RAM laptop computer. First, the plaintext and a secret key are identified. Then, the secret key is matched for every character in the plaintext and repeated based on the length of the message. Next, the ASCII decimal code value is retrieved based on each of the characters in the secret key. The ASCII values are used to perform the number of shifts in the matrix elements to generate the code equivalent for each plaintext character. For example, the message POSSESSION with the secret key CARL and its corresponding ASCII decimal codes is shown in Table XII.

The matrix is first shifted 67 cells to the right, as shown in Table XIII, to encrypt the first character of the plaintext. The process produces a new matrix wherein character P is encoded as the ciphertext 22, as presented in Table XIV.

Next, the matrix is shifted again by 65 times based on the ASCII equivalent of the following secret key character A. With the new matrix, the plaintext character O is encoded as 51, as shown in Tables XV and XVI.

TABLE XII. Plaintext and Key with ASCII Equivalent

Plaintext	P	O	S	S	E	S	S	I	O	N
Key	C	A	R	L	C	A	R	L	C	A
ASCII Value	67	65	82	76	67	65	82	76	67	65

TABLE XIII. New Matrix after 1^{ST} Cell Shift

	1	2	3	4	5
1	I/J	K	L	M	N
2	O	P	Q	R	S
3	T	U	V	W	X
4	Y	Z	A	B	C
5	D	E	F	G	H

TABLE XIV. 1^{ST} Character Encryption using EPS

Plaintext	P	O	S	S	E	S	S	I	O	N
Key	C	A	R	L	C	A	R	L	C	A
ASCII Value	67	65	82	76	67	65	82	76	67	65
Ciphertext	22									

TABLE XV. New Matrix after 2^{ND} Cell Shift

	1	2	3	4	5
1	T	U	V	W	X
2	Y	Z	A	B	C
3	D	E	F	G	H
4	I/J	K	L	M	N
5	O	P	Q	R	S

TABLE XVI. 2^{ND} Character Encryption using EPS

Plaintext	P	O	S	S	E	S	S	I	O	N
Key	C	A	R	L	C	A	R	L	C	A
ASCII Value	67	65	82	76	67	65	82	76	67	65
Ciphertext	22	51								

The process is executed repeatedly until the end of the plaintext length. After all of the matrix shifts, the final ciphertext value is now regarded as 22 51 22 23 32 35 52 34 31 55, as presented in Table XVII.

The comparison between the traditional Polybius Square and the modified Polybius Square encryption using the same plaintext is shown in Table XVIII. Based on the results, it is evident that the ciphertext is entirely different from the result of the original technique. Also, depicted in the table are the encrypted values of repeating characters O and S. As observed, the traditional method uses the same substitution values for similar symbols, such that characters O and S are always replaced by 34 and 43, respectively.

On the other hand, the modified method produces a more varied ciphertext value, even for identical plaintext characters. As observed, no two similar characters have the same ciphertext equivalent. Also, having the same ciphertext values do not necessarily equate to being similar plaintext character. This denotes that even if the ciphertext is the same, they may not have equivalent plaintext value making frequency analysis and decryption even more confusing and complicated.

The decryption process requires access to the ciphertext and the secret key. Each character in the key is matched with every digraph in the ciphertext. This process is repeated based on the length of the encrypted message. Next, the ASCII decimal code value is retrieved based on each of the characters used in the key. The ASCII codes are used to perform the number of shifts in the matrix elements to retrieve the code equivalent for the ciphertext digraphs. The process is executed repeatedly until all digraphs are converted to their respective plaintext values. Table XIX shows the encrypted message 22 51 22 23 32 35 52 34 31 55 with the secret key CARL and its corresponding ASCII decimal codes.

As another example, the plaintext MISSISSIPPI is encrypted using the traditional and modified Polybius Square. The results are shown in Table XX.

The results manifest obvious patterns for the ciphertext values generated using the traditional method wherein the repeating values 24 for I, 43 for S, and 35 for P are shown several times. However, looking closely at the ciphertext produced by the Enhanced Polybius Square, it is apparent that no two same plaintext characters are encrypted identically. For example, the character S was substituted with the values 51, 14, 52, and 22. Also, having similar ciphertext codes does not necessarily mean they have the same plaintext values, such that the ciphertext 15 was used to substitute characters I and P.

B. Evaluation Method

In order to assess the effectiveness and efficiency of the proposed method for longer texts, the execution time is evaluated, and the frequency analysis is performed. Both modified and unmodified methods were developed and tested using the aforementioned environment.

The frequency analysis is used to predict the value of the ciphertext based on how often a character or code appears [37]. In order to test the proposed scheme through frequency analysis, a sample plaintext is first encrypted and then supplied to an online tool [59] from the website Dcode. The encrypted message is subjected to a digraphs-digits-only analysis. The following text was used for testing:

"thiscourseaimstoprovideyouwithdetailedknowledgeofimportanttechnologiesandapplicationthatareusedintheinternetduetothebroadnatureofthisfieldthecoursecoversonlyselectedtopicsfocussingfirstonsomeadvancedtopicsininternettechnologiesegwirelesslansmobileinternetmulticastandthenaselectionofcurrentandnextgenerationapplicationsandservicesegppiptvvoip"

TABLE XVII. Encrypted Values using EPS

Plaintext	P	O	S	S	E	S	S	I	O	N
Key	C	A	R	L	C	A	R	L	C	A
ASCII Value	67	65	82	76	67	65	82	76	67	65
Ciphertext	22	51	22	23	32	35	52	34	31	55

TABLE XVIII. Comparison between Methods

Traditional Polybius Square										
Plaintext	P.	O.	S.	S.	E.	S.	S.	I.	O.	N.
Ciphertext	35	34	43	43	15	43	43	24	34	33
Enhanced Polybius Square										
Plaintext	P.	O.	S.	S.	E.	S.	S.	I.	O.	N.
Key	C	A	R	L	C	A	R	L	C	A
ASCII Value	67	65	82	76	67	65	82	76	67	65
Ciphertext	22	51	22	23	32	35	52	34	31	55

TABLE XIX. Decrypted values using EPS

Ciphertext	22	51	22	23	32	35	52	34	31	55
Key	C	A	R	L	C	A	R	L	C	A
ASCII Value	67	65	82	76	67	65	82	76	67	65
Plaintext	P.	O.	S.	S.	E.	S.	S.	I.	O.	N.

TABLE XX. Comparison using Repeated Letters

Traditional Polybius Square											
Plaintext	M	I	S	S	I	S	S	I	P	P	I
Ciphertext	32	24	43	43	24	43	43	24	35	35	24
Enhanced Polybius Square											
Plaintext	M	I	S	S	I	S	S	I	P	P	I
Key	U	S	U	S	U	S	U	S	U	S	U
ASCII Value	85	83	85	83	85	83	85	83	85	83	85
Ciphertext	52	12	51	14	15	52	22	21	52	15	24

Presented in Table XXI is the result of the frequency analysis of the encoded text using the traditional Polybius Square. Based on the result, the digraphs 15, 44, 24, 33, 34, 43, and 11 have the most count of repeating codes in the encrypted text. If these values are converted using the traditional method, the decrypted values would be E, T, I, N, O, S, A, respectively. The results adhere to the paper of [60] which discusses the top 7 most frequent letters in the Latin alphabet which are: E, A, T, I, O, N and S. This just means that a substitution cipher such as the Polybius Square is indeed prone to frequency analysis and therefore easy to break [2], [4], [34], [57].

Presented in Table XX is the result of the frequency analysis of the generated ciphertext using Enhanced Polybius Square. Based on the results, it is evident that there is minimal difference in the occurrence of each digraph. However, it can also be understood that decoding an EPS encoded message through frequency analysis may be very difficult and may take a long time since every digraphs or code can represent a number of plaintext values such that the digraphs 13 could mean any of the 25 letters depending on the series of transmutations on the matrix. Thus, making EPS not prone to frequency analysis.

TABLE XXI. Frequency Analysis results using the unmodified Polybius Square

Digraphs	**Frequency**	**%**
15	45	13.16
44	36	10.53
24	31	9.06
33	30	8.77
34	29	8.48
43	24	7.02
11	21	6.14
42	17	4.97
13	16	4.68
14	15	4.39
31	14	4.09
35	12	3.51
23	10	2.92
45	9	2.63
22	7	2.05
51	6	1.75
21	6	1.75
32	5	1.46
52	3	0.88
54	2	0.58
12	2	0.58
25	1	0.29
53	1	0.29

TABLE XXII. EPS FREQUENCY ANALYSIS RESULT

Digraphs	Frequency	%
13	21	6.14
45	20	5.85
54	20	5.85
31	18	5.26
15	18	5.26
23	17	4.97
11	17	4.97
25	15	4.39
35	15	4.39
51	15	4.39
22	13	3.8
14	13	3.8
24	13	3.8
21	13	3.8
41	12	3.51
43	12	3.51
33	12	3.51
34	12	3.51
42	12	3.51
52	12	3.51
12	11	3.22
55	8	2.34
44	8	2.34
52	8	2.34
32	7	2.05

With the extent to the processing time, it has been revealed that EPS has higher execution time with 0.0031s as compared to the unmodified method with 0.0005s. The results can be attributed to the fact that a new matrix is generated for every encrypted character against a static grid of the traditional Polybius Square. The simulation results are shown in Table XXIII.

With the use of the identified evaluation methods, the simulation results revealed that EPS performs better as the proposed method provides more layers of security through the use of a secret key and dynamically generated matrices.

TABLE XXIII. EXECUTION TIME INDEXED RESULTS

String:	MISSISSIPPI
Length:	11 characters
EPS Key:	US
Traditional P.S. Execution Time	0.003192900000000165 sec
EPS Execution Time	0.0005627999999999744 sec

V. CONCLUSION

This study presents a modification on the traditional Polybius Square through cell shifting and dynamic matrix generation using a keys' ASCII code. It has been revealed that the proposed method is more secure and is difficult to break via frequency analysis. However, the increase in security comes with a tradeoff in its execution time. It is recommended that further studies may be conducted to solve this problem.

REFERENCES

[1] S. Soomro, M. R. Belgaum, Z. Alansari, and R. Jain, "Review and open issues of cryptographic algorithms in cyber security," in International Conference on Computing, Electronics and Communications Engineering, iCCECE 2019, 2019, pp. 158–162.

[2] W. Stallings, Cryptography and Network Security Principles and Practices. Prentice Hall, 2015.

[3] B. N. Rao, D. Tejaswi, K. A. Varshini, K. P. Shankar, and B. Prasanth, "Design of modified AES algorithm for data security," Int. J. Technol. Res. Eng., vol. 4, no. 8, pp. 1289–1292, 2017.

[4] O. Reyad, "Cryptography and Data Security: An Introduction," 2018.

[5] S. N. Kumar, "Review on Network Security and Cryptography," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 8, no. 6, p. 21, 2018.

[6] D. Kahn, The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. Scribner, 1996.

[7] P. W. L. McLaren, "Investigations into Decrypting Live Secure Traffic in Virtual Environments," 2019.

[8] M. S. Hossain Biswas et al., "A systematic study on classical cryptographic cypher in order to design a smallest cipher," Int. J. Sci. Res. Publ., vol. 9, no. 12, pp. 507–11, 2019.

[9] M. Maxrizal and B. D. Aniska Prayanti, "Application of Rectangular Matrices: Affine Cipher Using Asymmetric Keys," CAUCHY –Jurnal Mat. Murni dan Apl., vol. 5, no. 4, pp. 181–185, 2019.

[10] T. M. Aung and N. N. Hla, "A Complex Polyalphabetic Cipher Technique Myanmar Polyalphabetic Cipher," in 2019 International Conference on Computer Communication and Informatics, ICCCI 2019, 2019, pp. 1–9.

[11] O. Laia, E. M. Zamzami, Sutarman, F. G. N. Larosa, and A. Gea, "Application of Linear Congruent Generator in Affine Cipher Algorithm to Produce Dynamic Encryption," J. Phys. Conf. Ser., vol. 1361, no. 1, pp. 1–6, 2019.

[12] J. C. Das and D. De, "Atbash cipher design for secure nanocommunication using QCA," Nanomater. Energy, vol. 6, no. 1, pp. 36–47, 2017.

[13] H. Nurdiyanto, R. Rahim, A. S. Ahmar, M. Syahril, M. Dahria, and H. Ahmad, "Secure a Transaction Activity with Base64 Algorithm and Word Auto Key Encryption Algorithm," J. Phys. Conf. Ser., vol. 1028, pp. 1–11, 2018.

[14] A. Abd and S. Al-Janabi, "Classification and Identification of Classical Cipher Type Using Artificial Neural Networks," J. Eng. Appl. Sci., vol. 14, no. 11, pp. 3549–3556, 2019.

[15] F. Anwar, E. H. Rachmawanto, C. A. Sari, and de Rosal Ignatius Moses Setiadi, "StegoCrypt Scheme using LSB-AES Base64," in International Conference on Information and Communications Technology, ICOIACT 2019, 2019, pp. 85–90.

[16] A. R. Pathak, S. Deshpande, and M. Panchal, "A Secure Framework for File Encryption Using Base64 Encoding," in Computing and Network Sustainability, vol. 75, Springer Singapore, 2019, pp. 359–366.

[17] R. Rahim, S. Sumarno, M. T. Multazam, S. Thamrin, and S. H. Sumantri, "Combination Base64 and GOST algorithm for security process," J. Phys. Conf. Ser., vol. 1402, 2019.

[18] 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, 2019, vol. 841, pp. 157–166.

[19] 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., vol. 1255, 2019.

[20] D. Gautam, C. Agrawal, P. Sharma, M. Mehta, and P. Saini, "An Enhanced Cipher Technique Using Vigenere and Modified Caesar Cipher," in 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018, 2018.

[21] J. Liu et al., "The Reincarnation of Grille Cipher: A Generative Approach," Cryptogr. Secur., pp. 1–27, 2018.

[22] P. E. Coggins and T. Glatzer, "An Algorithm for a Matrix-Based Enigma Encoder from a Variation of the Hill Cipher as an Application of 2 × 2 Matrices," Primus, vol. 30, no. 1, 2020.

[23] M. Shumay and G. Srivastava, "PixSel: Images as book cipher keys an efficient implementation using partial homophonic substitution ciphers," Int. J. Electron. Telecommun., vol. 64, no. 2, pp. 151–158, 2018.

[24] G. Zhong, "Cryptanalysis of Homophonic Substitution Cipher Using Hidden Markov Models," 2016.

[25] R. Deepthi, "A Survey Paper on Playfair Cipher and its Variants," Int. Res. J. Eng. Technol., vol. 4, no. 4, pp. 2607–2610, 2017.

[26] 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," in International Seminar: Research, Technology and Culture, 2017.

[27] R. Rahim and A. Ikhwan, "Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher," Int. J. Sci. Res. Sci. Technol., vol. 2, no. 6, pp. 71–78, 2016.

[28] H. B. Macit, A. Koyun, and M. E. Yüksel, "Embedding Data Crypted With Extended Shifting Polybius Square Supporting Turkish Character Set," BEU J. Sci., vol. 8, no. 1, pp. 234–242, 2019.

[29] 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., vol. 1361, no. 012020, 2019.

[30] G. Manikandan, P. Rajendiran, R. Balakrishnan, and S. Thangaselvan, "A Modified Polybius Square Based Approach for Enhancing Data Security," Int. J. Pure Appl. Math., vol. 119, no. 12, pp. 13317–13324, 2018.

[31] M. Maity, "A Modified Version of Polybius Cipher Using Magic Square and Western Music Notes," Int. J. Technol. Res. Eng., vol. 1, no. 10, pp. 1117–1119, 2014.

[32] C. Kumar, S. Dutta, and S. Chakraborty, "A Hybrid Polybius-Playfair Music Cipher A Hybrid Polybius-Playfair Music Cipher," Int. J. Multimed. Ubiquitous Eng., vol. 10, no. 8, pp. 187–198, 2015.

[33] 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, pp. 737–750.

[34] A. P. U. Siahaan, "Rail Fence Cryptography in Securing Information," Int. J. Sci. Eng. Res., vol. 7, no. 7, pp. 535–538, 2016.

[35] "'Unbreakable' Codes Throughout History: The Polybius Square and the Caesar Shift," 2011. [Online]. Available: https://freshmanmonroe.blogs.wm.edu/2011/07/17/"unbreakable"-codes-throughout-history-the-polybius-square-to-the-caesar-shift/.

[36] J. C. T. Arroyo, C. E. Dumdumaya, and A. J. P. Delima, "Polybius Square in Cryptography : A Brief Review of Literature," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 3, pp. 3798–3808, 2020.

[37] J. F. Dooley, History of Cryptography and Cryptanalysis. 2018.

[38] D. Salomon, Coding for Data and Computer Communication. Springer, 2005.

[39] I. B. Venkateswarlu and J. Kakarla, "Password security by encryption using an extended ADFGVX cipher," Int. J. Inf. Comput. Secur., vol. 11, no. 4–5, pp. 510–523, 2019.

[40] R. Mahendran and K. Mani, "Generation of Key Matrix for Hill Cipher Encryption Using Classical Cipher," 2nd World Congr. Comput. Commun. Technol. WCCCT 2017, pp. 51–54, 2017.

[41] G. Lasry, I. Niebel, N. Kopal, and A. Wacker, "Deciphering ADFGVX messages from the Eastern Front of World War I," Cryptologia, vol. 41, no. 2, pp. 101–136, 2017.

[42] A. Borodzhieva, "MATLAB-based software tool for implementation of Bifid Ciphers," in International Conference on Computer Systems and Technologies, 2017, pp. 326–333.

[43] E. V. Haryannto, M. Zulfadly, Daifiria, M. B. Akbar, and I. Lazuly, "Implementation of Nihilist Cipher Algorithm in Securing Text Data with Md5 Verification," J. Phys. Conf. Ser., vol. 1361, no. 012020, 2019.

[44] R. N. Sari, R. S. Hayati, Hardianto, A. H. Azhar, L. Sipahutar, and I. Lazuly, "Implementation of Trifid Cipher Algorithm in Securing Data," in 2019 7th International Conference on Cyber and IT Service Management, CITSM, 2019.

[45] M. Lavanya, E. Nixson, R. Vidhya, R. V. Sai, and K. Chakrapani, "Efficient data security algorithm using combined aes and railfence technique," Int. J. Pure Appl. Math., vol. 118, no. 20, pp. 3219–3227, 2018.

[46] G. Sharma, "Analysis and Implementation of DES Using FPGA," Thapar University, 2012.

[47] T. S. Kondo and L. J. Mselle, "An Extended Version of the Polybius Cipher," Int. J. Comput. Appl., vol. 79, no. 13, pp. 30–33, 2013.

[48] S. B. Olaleye and S. Ojha, "Improved Advanced Encryption Using Four Square Cipher for User Anonymity and Untraceability in Mobile Cloud Computing," Int. J. Innov. Sci. Eng. Technol., vol. 4, no. 2, pp. 113–121, 2017.

[49] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security," Optik (Stuttg)., vol. 127, no. 4, pp. 2341–2345, 2016.

[50] Z. Rahman, A. D. Corraya, M. A. Sumi, and A. N. Bahar, "A Novel Structure of Advance Encryption Standardwith 3-Dimensional Dynamic S-box and KeyGeneration Matrix," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 2, pp. 314–320, 2017.

[51] F. Patel and M. Farik, "A New Substitution Cipher - Random-X," Int. J. Sci. Technol. Res., vol. 5, no. 11, pp. 125–128, 2015.

[52] K. Al Harthy, F. Al Shuhaimi, and K. K. J. Al Ismaily, "The upcoming Blockchain adoption in Higher-education: Requirements and process," in 4th MEC International Conference on Big Data and Smart City, ICBDSC 2019, 2019, pp. 1–5.

[53] P. Kuppuswamy, R. Banu, and N. Rekha, "Preventing and securing data from cyber crime using new authentication method based on block cipher scheme," in 2nd International Conference on Anti-Cyber Crimes, ICACC 2017, 2017, pp. 113–117.

[54] R. Beck, M. Avital, M. Rossi, and J. B. Thatcher, "Blockchain Technology in Business and Information Systems Research," Bus. Inf. Syst. Eng., vol. 59, no. 6, pp. 381–384, 2017.

[55] S. Cho, Y. Jeong, and C. Oh, "An efficient cryptography for healthcare data in the cloud environment," J. Converg. Inf. Technol., vol. 8, no. 3, pp. 63–69, 2018.

[56] M. G. Vigliotti and H. Jones, "Cryptography for Busy People," in The Executive Guide to Blockchain, 2020, pp. 23–40.

[57] D. Kahn, Codebreakers. Macmillan and Sons, 1967.

[58] P. Kumar and S. B. Rana, "Development of Modified Polybius Technique for Data Security," Int. J. Innov. Eng. Technol., vol. 5, no. 2, pp. 227–229, 2015.

[59] "Frequency Analysis Tool," https://www.dcode.fr/frequency-analysis.

[60] G. Grigas and A. Juškevičienė, "Letter Frequency Analysis of Languages Using Latin Alphabet," Int. Linguist. Res., vol. 1, no. 1, pp. 18–31, 2018.