ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse117932020.pdf
https://doi.org/10.30534/ijatcse/2020/117932020

A Modified Nihilist Cipher Based on XOR Operation

Jan Carlo T. Arroyo[1], Allemar Jhone P. Delima[2]
[1]College of Computing Education, University of Mindanao, Davao City, Davao del Sur, Philippines
[2]College of Engineering, Technology and Management, Cebu Technological University-Barili Campus, Cebu, Philippines
jancarlo_arroyo@umindanao.edu.ph[1], allemarjpdjca@yahoo.com[2]

ABSTRACT

Encryption has become an essential component of most data security strategies. Most of all, organizations, under no circumstances, rely on basic authentication and access control systems to safeguard data. This paper presents a new method of ciphering messages using Nihilist cipher. The traditional way of ciphering, which is through the addition of both plaintext and key bigrams, is omitted. In the proposed technique, the bigram of the plain text and the key is XORed. The proposed method is applied to some known plaintext to test its effectivity. The use of the XOR-based Nihilist cipher on some identified plaintext samples generates secured, ubiquitous, and incomprehensible ciphertext, which is difficult to unravel as evident in the pattern analysis performed.

Key words: Cryptography, cipher, encryption, Nihilist cipher, XOR

1. INTRODUCTION

Hospitals, schools, government agencies, corporate companies, and enterprises - all of these organizations have crucial data on their desktop or in the drive [1]. In order to secure confidential data, cryptography is performed [2]. To provide security, files are converted into an unintelligible format. The art of hiding files is known as a cipher, where encryption and decryption of text, to name one, is observed [3].

Some of the commonly used ciphers that exist are Nihilist Cipher [4], Bifid cipher [5], Caesar Cipher [6]–[8], Homophonic Substitution Cipher [9], [10], and Polybius Square Cipher [14]–[17]. In this study, the traditional Nihilist Cipher, as one of the simplest cipher, is modified to address its drawback as to vulnerability in its key pattern. Since the length of the key can be guessed by inspecting for high and low number patterns, the cipher process is modified by introducing the XOR process.

2. EXISTING SYSTEM

2.1 Nihilist Cipher

The Nihilist cipher is a monoalphabetic classical cipher used by the Russians against Czar. The cipher was used as a medium of communication in prison by tapping the codes on cell walls [15]. The algorithm works the same as a Polybius cipher wherein a substitution is applied through the use of a matrix. The plaintext and the keyword are translated into its numerical form through substitution via a Polybius square to produce bigrams that represent the coordinates of the character in the grid. Each bigram generated from the plaintext and keyword is then summed to generate the ciphertext [16].

The standard Nihilist cipher uses the traditional 5x5 Polybius square grid matrix, as shown in Table 1. The matrix is filled with the Latin alphabets written from left to right and top to bottom.

Table 1: Nihilist cipher

	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

First, a plaintext value is translated to numeric value by matching each character to the given matrix and retrieving its row-column index known as a bigram. For example, the plaintext MESSAGE is converted into 32 15 43 43 11 22 15, as shown in Table 2.

Table 2: Plaintext conversion

Plaintext	M	E	S	S	A	G	E
Position	1	2	3	4	5	6	7
Converted Plaintext	32	15	43	43	11	22	15

Next, the keyword is also translated to its numeric equivalent by matching each character to the given matrix and retrieving its equivalent bigrams. For example, the keyword KEY is converted to 25 15 45, as shown in Table 3.

Table 3: Keyword conversion

Plaintext	K	E	Y
Position	1	2	3
Converted Keyword	25	15	45

Then, each character of the key is paired with each character of the plaintext. Since the given keyword is composed of only three letters, the characters are repeatedly matched up to the plaintext length, as seen in Table 4.

Table 4: Plaintext-keyword pairing

Plaintext	M	E	S	S	A	G	E
Converted Plaintext	32	15	43	43	11	22	15
Keyword	K	E	Y	K	E	Y	K
Converted Keyword	25	15	45	25	15	45	25

Lastly, to encrypt the plaintext using the keyword, the ciphertext equivalents are summed to generate the final ciphertext value. Based on the given, the first character M is encrypted as 57 (32+25), E is encrypted as 30 (15+15), and so forth. Therefore, the plaintext MESSAGE is encrypted as 57 30 88 68 26 67 40 using the keyword KEY, as seen in Table 5.

Table 5: Nihilist cipher encryption

Plaintext	M	E	S	S	A	G	E
Converted Plaintext	32	15	43	43	11	22	15
Keyword	K	E	Y	K	E	Y	K
Converted Keyword	25	15	45	25	15	45	25
Final Ciphertext	57	30	88	68	26	67	40

One advantage of using the Nihilist cipher over the Polybius cipher is that the former may generate varied ciphertext values for identical characters as opposed to the latter, which produces the same values for identical characters. For instance, the character S is encrypted as 88 or 68 and E as 30 or 40 based on the results above.

The decryption process requires access to the keyword. First, the given keyword is translated to its numerical equivalent. This is then deducted from the ciphertext to produce the bigrams for the plaintext. Each bigram for the plaintext is now matched against the Polybius square to retrieve the plaintext.

Cryptanalysis for Nihilist cipher is done through pattern analysis and factoring. The keyword length can be guessed by looking at low and high number patterns in the ciphertext. This is more obvious if a lengthy plaintext is used. The keyword may also be guessed by factoring, such that the ciphertext 89 can only be made by the factors 45+44 based on the limited combination of 1, 2, 3, 4, 5 from the matrix. This is due to the simple addition approach of the repeating keywords paired with the plaintext. For example, in a 5x5 matrix, if a ciphertext value is more than 100, then that would easily mean that both plaintext and keyword values come from the 5th row of the Polybius square.

3. PROPOSED METHODOLOGY

The proposed process introduces the use of an additional keyword and the XOR operation to further enhance the security of the cipher. The improvement in this study adheres to the recommendation of [17] on extending the Polybius square by incorporating a keyword in the generation of the matrix. Further, the XOR operation is executed to minimize patterns in the ciphertext values. The encryption process is presented in Figure 1, while the decryption process is presented in Figure 2.

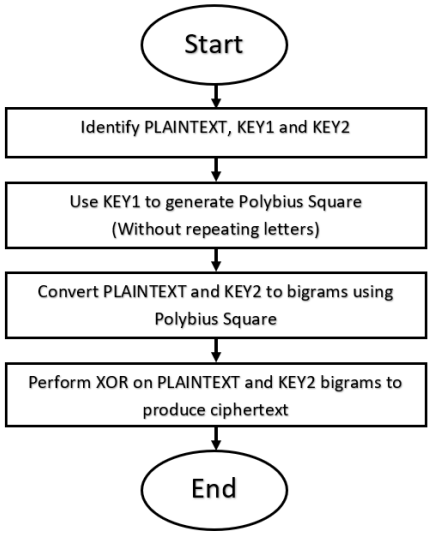

Figure 1: Encryption process

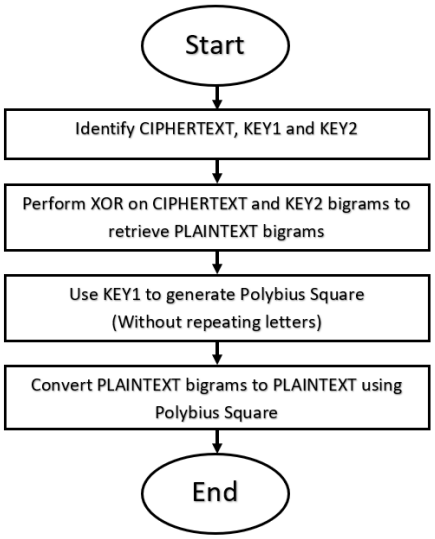

Figure 2: Decryption process

Encryption using the modified method requires a plaintext and two keywords. The first keyword is used to generate an extended 6x6 Polybius square composed of letters and digits. The keyword is added first before any other characters in the table. Any repeating characters are discarded. For instance, the first keyword CIPHER is plotted in the matrix, as shown in Table 6.

Table 6: Extended Polybius square with keyword

	1	2	3	4	5	6
1	C	I	P	H	E	R
2	A	B	D	F	G	J
3	K	L	M	N	O	Q
4	S	T	U	V	W	X
5	Y	Z	0	1	2	3
6	4	5	6	7	8	9

Next, the plaintext and the second keyword are translated into their numerical equivalent using the generated matrix. For example, the resulting conversion of the plaintext

MESSAGE, and the second keyword KEY is presented in Table 7. Each character from the key is paired with each character from the plaintext. This is done repeatedly until the last character of the plaintext is paired with a character of the key.

Table 7: Plaintext and keyword conversion

Plaintext	M	E	S	S	A	G	E
Converted Plaintext	32	15	43	43	11	22	15
Keyword	K	E	Y	K	E	Y	K
Converted Keyword	25	15	45	25	15	45	25
Final Ciphertext	57	30	88	68	26	67	40

Lastly, to encrypt the plaintext using the second keyword, the ciphertext equivalents are XORed to generate the final ciphertext value. Based on the given example, the first character M is encrypted as 62 (32 $\oplus$25), E is encrypted as 00 (15$\oplus$15), and so forth. With that, the plaintext MESSAGE will be encrypted as 57 30 88 68 26 67 40 using the keyword KEY, as seen in Table 8.

Table 8: Nihilist cipher encryption

Plaintext	M	E	S	S	A	G	E
Converted Plaintext	33	15	41	41	21	25	15
Keyword	K	E	Y	K	E	Y	K
Converted Keyword	31	15	51	31	15	51	31
Final Ciphertext	62	00	26	54	26	42	16

The decryption process requires access to the first and second keywords. The second keyword is translated to its numerical equivalent. This is then XORed from the ciphertext to produce the bigrams for the plaintext. Each bigram for the plaintext is now matched against the Polybius square, generated using the first keyword, to retrieve the plaintext.

4. RESULTS AND DISCUSSION

In order to assess the viability of the proposed method, two test cases were conducted using a variety of plaintext and keys. In this study, a 6x6 Polybius square was used to test both the standard and modified Nihilist ciphers. For the standard cipher, only keyword2 was used since it only needs one keyword. On the other hand, both keyword1 and keyword2 were used in testing the modified process.

The test case 1 shown in Table 9 presents a 43-bytes plaintext that is encrypted using the keywords ANIMAL and DOG. Each encrypted ciphertext for the standard and modified Nihilist cipher is checked for its low and high pattern compared to each next digit. Looking at the pattern results of the standard Nihilist cipher, a cryptanalyst may easily identify the pattern LOW HIGH LOW (d-o-g) because of how frequent it appears (6 times). Factoring may also be used to easily identify the keyword bigram in the standard Nihilist cipher, such that the first ciphertext bigram 27 can only be formed using the digits 11+16, 12+15, or 14+13. The modified method performs best as the LOW HIGH LOW pattern is not as obvious since it only appears twice. Most importantly, factoring cannot determine the keyword bigram because the Polybius square does not generate bigrams lower than 11 and higher than 66.

The test case 2 shown in Table 10 presents a 77-bytes plaintext that is encrypted using the keywords HIDDEN and INFO. Each encrypted ciphertext for the standard and modified Nihilist cipher is checked for its low and high pattern compared to each next digit. Looking at the pattern results of the standard Nihilist cipher, a cryptanalyst may easily identify the pattern LOW HIGH LOW HIGH (i-n-f-o) because of how frequent it appears (8 times). Factoring may also be used to identify the keyword bigram in the standard Nihilist cipher, such that the third ciphertext bigram 31 can only be formed using the digits 16+15. The modified method performs best as an obscure LOW HIGH LOW HIGH pattern is perceived since it only appeared five times. Most importantly, cryptanalysis using factorization cannot determine the keyword bigram because the Polybius square does not generate bigrams lower than 11 and higher than 66.

Table 9: Test case 1

Plaintext	cryptographyisthescienceofhidinginformation
Size	43 bytes
Keyword1	Animal
Keyword2	Dog
Nihilist Ciphertext	27 69 72 48 75 54 35 69 32 48 55 72 37 74 63 36 48 62 27 56 36 46 46 36 47 49 43 37 47 44 46 54 44 46 49 54 50 64 32 56 56 54 46
Pattern	Low High High Low High Low Low High Low High High High Low High Low Low High High Low High Low High Low Low High High Low Low High Low High High Low High High High Low High Low High Low Low Low
Modified Nihilist Ciphertext	03 05 42 52 11 56 15 05 18 52 59 42 27 08 51 12 54 48 03 44 14 26 52 14 55 57 03 27 55 20 26 56 20 26 57 56 50 47 18 60 44 56 26
	Low High High High Low High Low Low High High High Low Low Low High Low High Low Low High Low High High Low High High Low High High Low High High Low High High Low Low Low Low High Low High Low

Table 10: Test case 2

Plaintext	TheNihilistcipherisamonoalphabeticclassicalcipherusedby theRussiansagainstCzar
Size	77 bytes
Keyword1	Hidden
Keyword2	info
Nihilist Ciphertext	65 54 31 65 46 54 39 59 46 73 58 46 46 66 38 48 59 55 57 44 54 65 48 66 34 58 50 55 34 44 31 75 46 45 29 59 34 73 57 56 36 43 42 46 46 66 38 48 59 75 57 48 37 44 67 75 45 47 52 76 64 73 39 44 55 73 27 54 34 55 48 74 65 45 68 44 59
Pattern	Low Low Low High Low High Low High Low High Low Low Low High Low High High Low High Low High High Low High Low High Low High Low High Low High Low Low Low High Low High Low Low Low High Low High Low High High Low Low Low High High Low High High High Low High Low High Low High Low Low High Low High
Modified Nihilist Ciphertext	38 04 25 46 00 04 27 62 00 38 61 55 00 45 28 47 40 03 62 49 44 46 24 00 28 16 53 42 28 26 25 11 00 25 01 62 28 38 62 45 26 31 08 55 00 45 28 47 40 36 62 47 01 26 36 11 07 01 51 10 37 38 27 49 03 38 07 57 28 03 24 08 38 25 35 49 40
	Low Low High High Low High High High Low High High Low Low High Low High Low Low High Low Low High Low Low High Low High Low Low Low Low Low Low High Low High High Low Low High Low High High Low High Low High Low High Low High Low Low High Low High High Low High Low Low High Low High Low High High Low

5. CONCLUSION

In this paper, the use of the extended Polybius square is observed to strengthen the cipher capability of the Nihilist cipher as recommended in the study of [17]. Further, the use of the XOR process prior to the generation of ciphertext combines the secret key and plaintext bigram in such a way that the message is concealed without knowing both the message and the key. The proposed method shows the diversity and obscure frequency patterns for each cipher characters, making protected data difficult to crack.

REFERENCES

[1] A. Rayarapu, A. Saxena, N. V. Krishna, and D. Mundhra, "Securing Files Using AES Algorithm," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 3, pp. 433–435, 2013.

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

[3] 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, p. p9662, 2019, doi: 10.29322/ijsrp.9.12.2019.p9662.

[4] 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, doi: 10.1088/1742-6596/1361/1/012020.

[5] A. Borodzhieva, "MATLAB-based software tool for implementation of Bifid Ciphers," in *International Conference on Computer Systems and Technologies*, 2017, pp. 326–333. https://doi.org/10.1145/3134302.3134333

[6] 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, doi: 10.1007/978-981-13-2285-3.

[7] 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, doi: 10.1088/1742-6596/1255/1/012077.

[8] 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, doi: 10.1109/ICOEI.2018.8553910.

[9] 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, doi: 10.24425/119363.

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

[11] 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. https://doi.org/10.17798/bitlisfen.455126

[12] 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.

[13] W. Bengal, "a Modified Version of Polybius Cipher Using Magic Square and Western Music Notes," vol. 1, no. 10, pp. 1117–1119, 2014.

[14] 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, doi: 10.14257/ijmue.2015.10.8.19.

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

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

[17] 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, doi: 10.5120/13803-1836.