Strengthening The Vernam Cipher Algorithm Using Multilevel Encryption Techniques

Deborah G. Brosas, Ariel M. Sison, Ruji P. Medina

Abstract— Vernam Cipher, known as One Time Pad, was proved to be unbreakable; however, the keystream digits have to be completely random. Stream Ciphers address the issues by forgoing a degree of security by using a pseudorandom number generator, yet a known-plaintext attack is still a challenge when the key is used more than once due to a weak classical combiner XOR. With this, the study improved the algorithm using multilevel encryption techniques. The proposed algorithm evaluated with the use of Strict Avalanche Effect Criterion with average results of 81.48%. Moreover, the combined results of the Randomness Test having the P-Value between 0-1 (Table 4) indicates that the proposed method showed better results over the original Vernam Cipher algorithm.

Index Terms— Avalanche Effect, Cryptography, Multilevel Encryption, One Time Pad, Statistical Test, Stream Cipher, Vernam Cipher

1 INTRODUCTION

Acryptographic system is supposed to be secure if the ciphertext does not contain adequate details to find out the plaintext[1]. There are many mechanisms in cryptography science to offer solutions for various security cases such as Vernam cipher, mono-alphabetic cipher, and poly-alphabetic cipher [2]. Most of these are to develop or discover techniques that need randomness for many different reasons [3]. The two essential characteristics of cryptography are the strength of the encryption algorithm and the secrecy of the key [4]. Respectable papers have been proposed by [5], [6] and [7] with the idea of directing a goal of preserving the secret text from brute force attack by enhancing the substitution process with ASCII code which is significant in computer systems.

Literatures showed that as of today, Vernam Cipher known as One-Time-Pad or OTP, is still famous for its perfect secrecy. Shannon proved that Vernam cipher is secure. According to [8]and [9], Ciphertext (C) is random and independent of Message (M) when Key (K) is random and unknown except for the key, length because it requires a key as long as the plaintext.

An enhanced key or pad generation process of Vernam Cipher was proposed by [10], [11], [12], and [13] with the use of pseudorandom random number generators or PRNG. An improved version of the said cipher was focused mostly on the Key generation process and extending the random keys of an OTP. The essential issues in the said cipher were when a key used more than once it becomes vulnerable to a pattern analysis attack or known-plaintext attack [10][14].

According to [15] and [11], a known-plaintext attack is the key challenge faced in Vernam Cipher if the key is processed more than once due to a weak classical combiner XOR. It is due to enough redundancy in English and ASCII encoding a good pattern analysis or crib dragging help the eavesdropper reach either or both the Plaintexts[10]. Therefore, the study provides an improved encryption scheme with the use of multilevel encryption techniques and combined substitution technique by way of the OTP technique. The strength of the cipher of the proposed algorithm has been tested using the Strict Avalanche Effect Criterion or (SAC).

- *Deborah G. Brosas is currently pursuing Doctorate degree program in Information Technology in Technological Institute of the Philippines, Quezon City, Philippines. E-mail: deb.brosas@outlook.com*
- *Ariel M. Sison is currently connected at School of Computer Studies in Emilio Aguinaldo College, Manila, Philippines. E-mail: ariel.sison@eac.edu.ph*
- *Ruji P. Medina is currently the Dean of Graduate Programs in Technological Institute of the Philippines, Quezon City, Philippines. E-mail: ruji.medina@tip.edu.ph*

2 RELATED LITERATURE OF THE STUDY

2.1 Modified Substitution Technique

In cryptography, the substitution technique is a method of encrypting the plaintext by swapping each character by different symbol as directed by the key. S. J. Manowar et al. [16], have modified the standard Vernam Cipher Method for all characters (ASCII code 0-255) with a randomized keypad. The proposed method can provide security against a passive attack, such as a password-based attack. According to [17], Ayussi (2010) has proposed an encryption algorithm based on concepts of ASCII values and bit reversal, but it faces disadvantages of using a fixed-length key and encrypting only small amount of data. Series of modifications were proposed to improve the process, one of which is a paper presented by [18], the first level of encryption was the algorithm encodes each character into ASCII codes. Also, according to the nucleotide sequence, the researcher should convert it to the DNA coding.

2.2 Modified Encryption Scheme

A polyVernam cipher proposed and modified by [19] to work with the keys, which were very complicated and unpractical to use. When combined with modern polyalphabetic cipher algorithms, it is possible to encrypt data by using the Key generation algorithm procedures and can also be repeatedly used as a simple password.

A new algorithm based on Vernam Cipher modified with an addition 2's complement to increase the complexity of the algorithm was proposed by [5] and [6]. The used key being optimization using a genetic approach to improve the strength and complexity from Key determination, but both studies

presented a low Avalanche Effect or AE score.

An enhanced practical difficulty of OTP algorithm was proposed by [10], the methodology was a hybrid of some cryptographic primitives to introduce diffusion and a simple randomized form of steganography to hide where the encryption begins. The result of the enhanced OTP algorithm was suitable to allow limitless use of the same key for encrypting different Plaintext without revealing any pattern, but the long Key encryption process in OTP needs improvement.

3 Proposed Modification Methods of the Vernam Cipher

The proposed modified algorithm is a software stream cipher that takes a Plaintext and a Key of up to 256 bits as input. The generated secret Key should be equal to the length of the plaintext. A generated random bit named as a Seed value be utilized to encipher the Key. The Seed is up to 256 bits or less depending on the size of the key generated.

The following are the modified interpretation of the Vernam cipher. The procedures of the different phases presented with different practices and technologies adopted to solve the problem addressed by the study. The proposed modification provides a multi-level encryption procedure.

In the proposed algorithm, the modified encryption scheme of the proposed algorithm involves three stages:

3.1 Key Generation

Nowadays, harvesting the entropy from different sources available within personal computers is an increasingly popular approach. The most well-known examples include the Linux random number generator accessible through /dev/random, the portable random number infrastructure EGADS – Entropy Gathering and Distribution System or the EGD – Entropy Gathering Daemon [20].

The generator mentioned above was used to generate the Key and random bit called Seed. Each generation can create characters of up to 256 in length.

The Seed Setup process is the following:

Step	Process
1	Start
2	Read the Key.
3	Read IV or Initialization Vector extracted from the used generator.
4	The conversion of character input to ASCII value then to equivalent binary form.
5	Enciphering characters using the proposed equation:

$$S_n = S_{n-1} \oplus (IV_{n+1} \oplus K_{n+1}) \bmod 256 \quad (1)$$

6	Converting the resulted binary numbers to the equivalent ASCII value to obtain the cipher characters called the Seed value.

3.2 Encryption Process

The proposed algorithm complicates the weak XOR combiner by adding more combining functions to the algorithm with a randomized keystream and the high avalanche effect result.

3.2.1 Key Encryption

The Output Feedback (OFB) technique was used to encipher the given Key with the given Seed value, where each bit of the Key is being XORed from the Seed value, and the resulted bit is XORed again to the next bits. The process is repeated up to the last bit. The output bits are named Encrypted Key Output 1. The following procedure is to use the circular shift technique to encipher *CT1,* and the output is the Encrypted Key Output 2 or *CT2*.

Fig. 1 shows the diagram of the key-encryption process of the proposed algorithm:

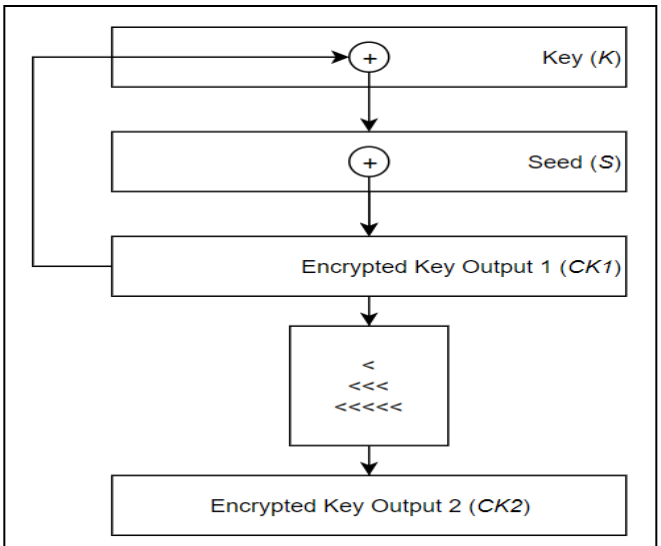

Fig. 1. Key encryption process of the proposed method.

Below is the sequential interpretation of the Key-encryption process.

Step	Process
1	Convert the Key and the Seed value to its binary equivalent.
2	Perform the equation below for encryption.

$$CK_n = CK_{n-1} \oplus (S_{n+1} \oplus K_{n+1}) \bmod 256 \quad (2)$$

3	Store the Encrypted Key Output1 as (CK1). Determine the Encrypted Key Output2 as (CK2) by executing left circular shift function using the equation below.

$$CK1_0 = CK1 < 1$$
$$CK1_1 = CK1_0 <<< 3$$
$$CK1_2 = CK1_1 <<<<< 5$$
$$CK2 = CK1_2 \quad (3)$$

3.2.2 Message Encryption

Fig. 2 shows the message encryption technique of the proposed algorithm. Encrypting the plaintext involves several stages. First is the Substitution process, and the next level is the encryption process, as shown below.

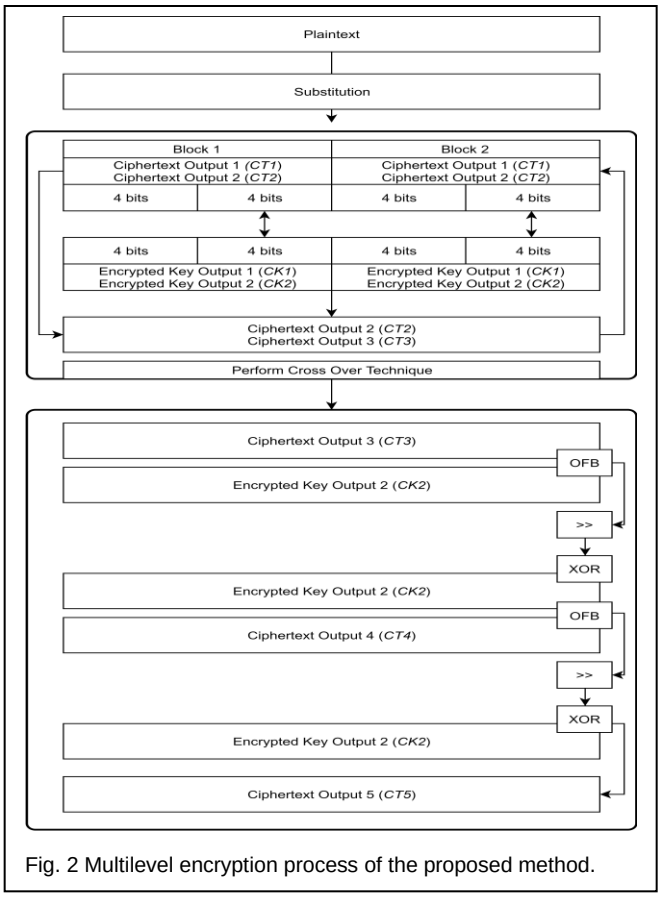

Fig. 2 Multilevel encryption process of the proposed method.

The following is the illustration of the step by step Message encryption process.

Substitution Process:

Step	Process
1	Start
2	Read the Plaintext
3	Convert Plaintext to ASCII value and obtain the binary sequence.
4	Convert binary values to DNA Binary Sequence using DNA Digital Coding Table.
5	Store Cipher Text Output 1 or CT1.

Encryption Process:

Step	Process
1	Start
2	Read Ciphertext Output 1 (CT1) and Encrypted Key Output 2 (CK2).
3	Execute Cross Over technique with Output Feedback. Store the obtained value to Ciphertext Output 2 (CT2).
4	Repeat Step 2 and 3 and store obtained value to Ciphertext Output 3 (CT3).
5	In getting Ciphertext Output 4 (CT4), the CT3 and CK2 are processed using Output Feedback operation. The resulted ciphertext value is shifted twice using a left circular shift. The result again is XORed to CK2. The resulted ciphertext stored on CT4.
6	Repeat step 5 by processing CT4 and CK2. Store the result on Ciphertext Output 5 (CT5).

3.3 Decryption Process

Step	Process
1	Start
2	Read Ciphertext Output 5 (CT5).
3	Read Encrypted Key Output 2 (CK2).
4	Reverse the last encryption, Step 6 process to obtain CT4.
5	To obtain CT3, read CT4, then reverse encryption Step 5.
6	Read CT2 and CK2, perform reverse crossover technique to obtain the value of CT1 and CK1.
7	Perform Reverse Substitution using DNA Digital Coding Table to obtain the Plaintext ASCII Value.
8	Obtain the Character value to display the original plaintext.

3.4 Strict Avalanche Criterion

In achieving a secure cipher, nothing can learn about cipher's behavior [9]. Testing for the randomness of the output ciphertext was evaluated using the Strict Avalanche Effect Criterion. The avalanche effect presents an intuitive idea of high-nonlinearity: a small difference in the input, making a massive transformation in the output, thus an avalanche of changes.

The Strict Avalanche Effect Criterion computed as follows:

$$\forall x,\ y | H(x,\ y) = 1, \quad \text{average}(H(F(x),\ F(y))) = \frac{n}{2} \tag{4}$$

3.5 Randomness Test

The cryptosystem's security is vastly related to the quality of randomness of the used technique or mechanism to encrypt a message or plaintext. Enciphering is a process used to encrypt a word based on the cryptographic random numbers. There are various statistical randomness tests introduced to determine whether a cryptographic random number generator is suitable for cryptographic applications or not.

In the proposed method, the randomness of the output ciphertext sequence is tested using the frequency (Monobit) test. In this test, it counts the number of 1s and 0s in the given bit sequence and checks the difference between bits. The other one is the runs test, these test targets to test if the number of runs (1 or 0) of indefinite length in the sequence is random.

4 Results and Discussions

The illustrations below are the sequential interpretation of the process of the proposed method.

4.1 Key Encryption

In the proposed Key encryption technique, a random bit or Seed is being used to encipher the Key. This process was included to allow multiple messages encryption with the same Key as shown in Table 1.

TABLE 1
KEY ENCRYPTION RESULT

Key	TheSecret103_256
Seed	Lter5_YgpVO_#tHP
Step 1: Perform Eq. 2	00010000001010000011111111111110011000000010100000 11001000111100001110000100010111010101001001000010 01111011101110100110111011
	00010000001010000011111111111110011000000010100000 11001000111100001110000100010111010101001001000010 01111011101110100110111011
Step 2: Perform Eq. 3	01100010000001010000011111111111110011000000010100 00011001000111100001110000100010111010101001001000 10101111011101110101001101111
	10111011000100000010100000111111111111100110000000 10100000110010001111000011100001000101101010100100 10000101011110111011101010011
CK2	» (?þ`(2<8EªHW»©

4.2 Message Encryption

In the proposed encryption scheme, the plaintext undergoes a multilevel encryption process, as presented in Table 2 to strengthened the random keystream generated or ciphertext.

TABLE 2
PLAINTEXT ENCRYPTION RESULT

SUBSTITUTION	
Message	**DementiaStroke_1**
ASCII	01000100011001010111011010110010101101110011101000 11010010110000101010011011101000111001001101111101 10101101100101010111110011 0001
DNA Sequence	CACGAGCGATCAAGAG
ASCII (CT1)	01000011010000010100001101000111010000010100011110 10000110100011101000001010101000100001101000001010 00000101000111010000010100 0111
ENCRYPTION	
CT1	01000011010000010100001101000111010000010100011110 10000110100011101000001010101000100001101000001010 00000101000111010000010100 0111
CK1	00010000001010000011111111111110011000000010100000 01100100011110000111000010001011010101001001000010 01011110111011101010011011 1011
Step 3: Perform Crossover with OFB operation	Ciphertext Output 2 (CT2) 01010011011010010111110010111001001000010110111110 11100010111101101111001000100011110100100000100100 01011011111100111010001111 1100

Table II Continuation

	01010011011010010111110010111001001000010110 11110111000101111011011100100010001111010 01
CT2	00001001000101101111110011101000011111100 10111011000100000010100000111111111111100110
CK2	00000010100000110010001111000011100001000101 10101010010010000101011110111011010100 1
Repeat Step 3: Perform Crossover with OFB operation	Ciphertext Output 3 (CT3) 11101000011110010101010010000110110111110000 11110101100101001001010001010010100110101100 10100011010111101010101101010011010101 01
CT3	11101000011110010101010010000110110111110000 11110101100101001001010001010010100110101100 10100011010111101010101101010011010101 01
CK2	10111011000100000010100000111111111111100110 00000010100000110010001111000011100001000101 10101010010010000101011110111011010100 1
Execute Step 5	Ciphertext Output 4 (CT4) 01000010111111010010100101011000010111101010 00110010101011000100110011111000000000011011 11001110100010110000001110110000001010111
Repeat Step 5	01010011010011111000110100010100111010110 11 01000100110101101010101001011100100001000110 11010011100110010011011111001101101000 11
Final Ciphertext	Ciphertext Output 5 (CT5) **SO□ □ õ´Mj¥ÈFÓ™7Íɛ**

4.3 Avalanche Effect

The proposed algorithm was compared and analyzed with the original and modified Vernam Cipher published last 2016 and 2019. The evaluation of Avalanche Effects based on different conditions was evaluated and presented in Table 3. The plaintext used in the experiment tested on different positions and a one-bit change in every simulation – the results as shown.

TABLE 3
AVALANCHE EFFECT PERFORMANCE

Plaintext Position Modification	**Avalanche Effect**			
	Vernam Cipher	**Modified Vernam Cipher (2016)**	**Modified Vernam Cipher (2019)**	**Modified Vernam with Multi-level Encryption**
Beginning	4.69	4.69	4.69	94.43
Middle	4.69	12.50	12.50	50.78
End	3.13	6.25	6.25	99.22
Average	4.17	7.81	7.81	**81.48**

The results conferred that the proposed algorithm always gets the highest Avalanche Effect. Average shows that in every bit change in the input, there is an average of 81.48% change in the output. Every part of the input affects every part of the output, which makes analysis become harder. It proved that it is more secure than the original algorithm.

4.5 Randomness Test Results

Table 2 below presented the overall comparison of the different variants of Vernam Cipher algorithms in terms of their P-Value results. A P-Value is a number between 0 and 1 and is used to weigh the strength of the evidence. Proportions of sequences passing a given test are the p-values higher than the significance level $\alpha = 0.01$ [21]. In every test, it shows that the P-Value of the proposed algorithm shows a promising result when comparing with all the other algorithms.

TABLE 4
RANDOMNESS TEST RESULT

Cryptosystem	P-Value Results			
	Frequency Monobit Test	Frequency Block Test	Runs Test	Remarks
Vernam Cipher	0.2113	0.13025	0.40947	Passed
Modified Vernam Cipher (2016)	0.61708	0.64723	0.07374	Passed
Modified Vernam Cipher (2019)	0.31731	0.64723	0.05681	Passed
Modified Vernam with Multi-level Encryption (Proposed Method)	**0.61708**	**0.98101**	**0.30053**	**Passed**

Figure 3 is a graphical representation of the randomness analysis of output of the compared algorithms. Based on the P-Value of each method, the test results are not small than 0.01, which is the default significance interval. Since the results indicated that the P-Value in every randomness test reflected in Table 4 is higher than the significant level, which is 0.01, the sequence is considered random.

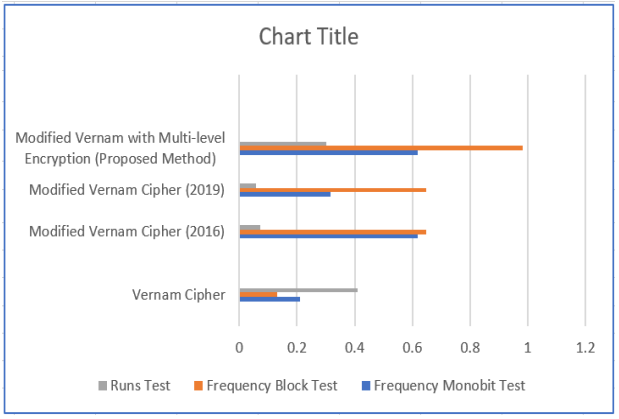

Fig. 3 Randomness test results.

The frequency block test result of the proposed method achieved the highest P-Value, which is 0.98, followed by the frequency mono bit test and the runs test.

5 Conclusion and Future Work

The results clearly show that based on the randomness statistical analysis of output of the proposed cryptosystem, the method is suitable for use in the encryption process. The combined results of the Randomness Test having the P-Value between 0-1 (Table IV) and an average avalanche result of 81.48 indicates that the proposed method strengthened the cipher against pattern analysis or crib dragging. For future works, the researcher would like to explore other substitution technique like using S-boxes to improve the proposed algorithm.

Acknowledgment

The authors are thankful for the understanding and knowledge conferred by the Almighty God in making this study possible in spite of every challenge.

References

[1]. P. Penchalaiah and K. R. Reddy, "Random multiple key streams for encryption with added CBC mode of operation ☪," Perspect. Sci., vol. 8, pp. 57–60, 2016.

[2]. A. Saraswat, C. Khatri, P. Thakral, and P. Biswas, "An Extended Hybridization of Vigenere and Caesar Cipher Techniques for Secure Communication," Procedia - Procedia Comput. Sci., vol. 92, pp. 355–360, 2016.

[3]. O. Salhab, N. Jweihan, M. A. B. U. Jodeh, and M. A. B. U. Taha, "SURVEY PAPER : PSEUDO RANDOM NUMBER," vol. 96, no. 7, 2018.

[4]. E. Edition, E. Edition, O. Systems, S. Edition, B. D. Communications, and S. Edition, THE WILLIAM STALLINGS BOOKS ON COMPUTER DATA AND COMPUTER COMMUNICATIONS, EIGHTH EDITION...□

[5]. B. K. Pawar, "Vernam Cipher : VSA," 2016 Int. Conf. Inven. Comput. Technol.

[6]. A. Rahman Dalimunthe, H. Mawengkang, S. Suwilo, and A. Nazam, "Vernam Cipher with Complement Method and Optimization Key with Genetic Algorithm," J. Phys. Conf. Ser., vol. 1235, p. 012030, 2019.

[7]. S. S. Kadhim, "Cryptography Using Modified Vernam - Homophonic Method Implemented by Matlab," pp. 539–550, 2017.

[8]. Y. Heights, "Quantum Vernam cipher," vol. 1, no. 0, pp. 1–21, 2001.

[9]. J. Aumasson, Serious Cryptography. No Starch Press, Inc.

[10]. A. E. Omolara, A. Jantan, O. I. Abiodun, and H. Arshad, "An Enhanced Practical Difficulty of One-Time Pad Algorithm Resolving the Key Management and Distribution Problem," vol. I, 2018.

[11]. G. Upadhyay and M. J. Nene, "One Time Pad Generation Using Quantum Superposition States," no. 1, pp. 1882–1886, 2016.

[12]. M. Devipriya and G. Sasikala, "A New Technique for One Time Pad Security Scheme with Complement Method," vol. 5, no. 6, pp. 220–223, 2015.

[13]. A. Omotunde and E. Onuiri, "An Implementation of a One-Time Pad Encryption Algorithm for Data Security in Cloud Computing Environment," no. December 2017.

[14]. P. Jattke, M. Senker, and A. Wiesmaier, "Comparison of two Lightweight Stream Ciphers :"
[15]. A. H. Kashmar, E. S. Ismail, F. M. Hamzah, and H. F. A. Amir, "Design a Secure Hybrid Stream Cipher," vol. 5, no. 3, pp. 153–166, 2015.
[16]. S. J. Manowar and A. M. Sahu, "Introduction to Modern Encryption Standard (MES) -II : An independent and efficient Cryptographic approach for Data Security," vol. 5, no. 1, pp. 310–313, 2014.
[17]. M. Lavanya, R. V. Sai, A. Festina, and J. Eshwari, "An Encryption Algorithm Functioning on ASCII Values and Random Number Generation," vol. 8, no. December, pp. 8–11, 2015.
[18]. Y. Zhang, X. Liu, and M. Sun, "DNA based Random Key Generation Management for OTP Encryption," BioSystems, 2017.
[19]. R. Divya and A. M. M, "The PolyVernam Cipher," vol. 3, no. Viii, pp. 489–496, 2015.
[20]. T. H. E. P. House, O. F. The, and R. Academy, "Generation and testing of random numbers for cryptographic applications *," vol. 13, no. 4, pp. 368–377, 2012.
[21]. M. Sýs, Z. Říha, V. Matyáš, K. Márton, and A. Suciu, "On the interpretation of results from the NIST statistical test suite," Rom. J. Inf. Sci. Technol., vol. 18, no. 1, pp. 18–32, 2015.