Symbiosis

www.symbiosisonlinepublishing.com
ISSN Online: 2474-9257

Research Article | Journal of Computer Science Applications and Information Technology | Open Access

A Two-Phase Symmetric Key Block Cipher

Eun-Joo Lee[1], Bose Omo-Ekpadi[1], and Haklin Kimm[1*]

*[*1]Department of Computer Science, East Stroudsburg University of Pennsylvania, 200 Prospect Street, East Stroudsburg, PA 18301-2999, USA*

Received: January 24, 2019; Accepted: February 13, 2019; Published: February 26, 2019

***Corresponding author:** Haklin Kimm, Department of Computer Science, East Stroudsburg University of Pennsylvania, 200 Prospect Street, East Stroudsburg, PA 18301-2999, USA, Email: HKimm@po-box.esu.edu*

Abstract

Improving the security of cipher text is an important issue in cryptography. We propose a two-phase symmetric-key block cipher to provide the enhanced security of cipher texts. In the first phase, Modified Symmetric Key Block Cipher (MSKBC) comprises 128-bit plaintext as a block and produces a cipher text which contains a quotient and a remainder from dividing the plaintext by the 52-bit key. A permutation cipher is utilized in the second-phase to re-arrange the cipher text obtained from the first-phase. Experimental results show that the proposed algorithm achieves confusion and diffusion, and exhibits significant high Avalanche Effect which improves the level of the security.

Key words: Symmetric key cipher; Block cipher; Confusion; Diffusion; Avalanche Effect;

Introduction

The explosive growth of computer systems has led to a heightened awareness of security. Protecting sensitive data from all kind of virtual threats is one important function of cryptography and computer security.

Cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages [1]. In cryptography, a cipher is an algorithm for performing encryption or decryption. Symmetric key cryptography refers to encryption/decryption methods in which both sender and receiver share the same key. Symmetric key ciphers are implemented as either block cipher or stream cipher. "A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length [2]".

According to Stallings, there are two requirements for secure use of symmetric key encryption; (1) a strong encryption algorithm and (2) obtaining the secret key in a secure fashion and must keep the key secure [2]. In addition, "the desired property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the cipher text. In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the cipher text. This is referred to as the avalanche effect. If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched."

In modern block encryption, it is generally considered that diffusion and confusion are the desired attributes to hinder statistical analysis [3]. The terms diffusion and confusion were introduced by Claude Shannon. The mechanism of diffusion seeks to make the statistical relationship between the plaintext and cipher text as complex as possible in order to thwart attempts to deduce the key. On the other hand, confusion makes the relationship between the cipher text and key as possible as complex, in order to make it difficult to deduce the key [4].

In 2010, Ayushi proposed New Symmetric Key Algorithm (NSKA) which utilizes a simple mathematical division method [5]. Based on the paper, NSKA is simple in nature and works very smoothly for a small amount of data. However, based on our research, NSKA could have several problems such as allowing only limited plaintext size of four characters, encryption one letter at one time, and small key size of four which cause weaken to a brute-force attack [5].

As part of a continuous effort on developing the security, we develop a symmetric block cipher algorithm based on a simple mathematical division method, closely related to the New Symmetric Key Algorithm [5]. The proposed two-phase algorithm encrypts a 16-bit letter (128-bit plaintext in binary form) as a block with 52-bit key size and produces a 33-bit cipher text in hexadecimal form for any plaintext size up to 16 letters. Experimental results show that altering one character in a plaintext produces 70 – 90 % of cipher text characters are altered with new characters. Also, changing a bit in the 52-bit key alters 70 – 78 % of cipher text characters.

This paper is organized as follows. Section 2 describes the New Symmetric Key Algorithm (NSKA) [5] as a background of our proposed algorithm. We introduce and justify the proposed algorithms in Section 3. In section 4, numerical results are presented to demonstrate the advantages and performance of the proposed algorithm. Concluding remarks are given in Section 5.

New Symmetric Key Algorithm

In this section, we first review the New Symmetric Key Algorithm by Ayushi [5].

Algorithm 2.1

Encryption: New Symmetric Key Algorithm (NSKA) [5]

1. Generate the ASCII value of the letter
2. Generate the corresponding binary of it
3. Reverse the 8 digit's binary number
4. Take a 4-digit divisor (>=1000) as the key
5. Divide the reversed number with the divisor
6. Store the remainder in the first 3 digits & quotient in next 5 digits
7. Store the remainder in the first 3 digits and quotient in next 5 digits

NSKA [5] encrypts a plaintext "T" and produces a cipher text "E". Specifically, (1) the ASCII of character "T" is 84 in decimal. (2) 84 s converted to 8-bit binary number 01010100 (3) Reverse this binary number would be 00101010 (4) Let 1000 as divisor i.e., Key (5) Divide 00101010 (divided) by 1000 (Key) (6) the remainder would be 10 and the quotient would be 101. So as per the algorithm, the cipher text would be 01000101 which is ASCII 69 in decimal i.e., "E."

Based on our research, however, NSKA [5] could have several problems: (1) it only allows limited plaintext size. Even the author says it is a block cipher; it encrypts a plaintext only one character in each step. (2) Limited key size. Due to the key size of 4, it has only 15 different key combinations from 0001 to 1111. So, we can say that the algorithm is very weak to a brute-force attack. (3) For some cases of keys, it produces the same plaintext. For example, Key = 1000, plaintext N, Q is the same as its cipher text. If Key = 1000, plaintext D, J, N, and Q are same as its plaintexts. If Key = 1010, plaintext D, J, N, Q, U are same as cipher text of those. (4) The algorithm is broken if the quotient is greater than 31 (11111_2) or the remainder is greater than 7 (111_2).

A Two-Phase Symmetric Key Block Cipher

We now introduce our two-phase symmetric key encryption algorithm based on a mathematical division method [5]. In the first phase, our Modified New Symmetric Key Algorithm (MNSKA) encrypts up to 16 characters of plaintext as a block and produces 32 characters of cipher text. In the second phase, we employ the double transposition cipher to improve the security of the cipher text after utilizing the MNSKA. The double transposition cipher is a columnar transposition applied twice. In this phase, two different keys are used for both transpositions.

Algorithm 3.1

Encryption: Modified New Symmetric Key Algorithm (MNSKA)

1. Generate 52-bit key K
2. Pad plaintext (P) with extra characters to a maximum size of 16 if the size of P is less than 16
3. Convert the plaintext P to 128-bit binary number
4. Compute Quotient Q and remainder R by P/K
5. Convert Q and R to hexadecimal
6. Cipher text C = Q + R

In the above algorithm, the Linear Feedback Shift Register (LFSR) is utilized to generates 52-bit key K with 10-bit initial vector, and the maximum number of bits can be generated for our key is 2^{10} - 1 = 1023[6, 7] .

Algorithm 3.1 describes the encryption process of MNSKA (Modified New Symmetric Algorithm). The algorithm encrypts 16 characters (plaintext P) as a block and produces a 33-bit cipher text C contains the quotient and remainder in hexadecimal by dividing the plaintext P by the key K. Plaintext P is padded with random characters to increase its length of 16 if the length of the plaintext is less than 16. For example, if the length of a plaintext P is 11, 5 extra characters padded to the plaintext. On computing the quotient Q and the remainder R, MPIR Library [7] is utilized due to the too large values to calculate with the regular C++ mathematical operators. For the decryption, the following algorithm 3.2. Shows the decryption process of MNSKA.

Algorithm 3.2

Decryption: Modified New Symmetric Key Algorithm (MNSKA)

1. Extract Quotient Q and Remainder R from the cipher text C
2. Convert the quotient Q and the remainder R from Hexadecimal to Decimal
3. Convert the key K to decimal
4. Compute the plaintext P multiplying by the key K and adding the remainder R to it, P = Q * K + R
5. Convert the plaintext from decimal to Binary and binary to ASCII
6. Remove the padding

The encryption of our two-phase symmetric key algorithm starts with MNSKA followed by the double transposition algorithm. In the first phase, MNSKA encrypts plaintext to cipher text1, and then the ciphtertext1 is encrypted again using the double transposition cipher utilizes the cipher text1 from the first phase and generates cipher text2. On the contrary, the decryption begins with phase 2. Cipher text2 is the input of the double transposition algorithm and it returns plaintex1, and then MKSKA decryption algorithm returns plaintext2 which is the same plaintext as the original plaintext.

Experimental Results

We present numerical experiments of the proposed encryption algorithm in this section. The key generation algorithm is developed based on the Linear Feedback Shift Register (LFSR) [6], and the key size used in this paper is 52-bit with the 10-bit initial vector. However, the algorithm could generate up to a 99-bit key for increasing the complexity of encryption exponentially. In addition, generating keys requires an only a small amount of

time using the algorithm. Based on our experiment, the times to generate 10 different keys are range from a low of 0.0015 seconds to a high of 0.0026 per each key generating event. Also, Multi-Precision Integers and Rational (MPIR) Library [7] is utilized to generate and handle very large numbers for the implementations. In all tables with the numerical results, we use the 52-bit key, 01 10100101101101101101101101101101101101101101101101, for our algorithm and 4-bit key, 1000 for NSKA [5].

Comparisons of Modified New Symmetric Key Algorithm (MNSKA) and NSKA

Table 4.1 shows the comparison between the proposed algorithm (MNSKA) and NSKA [5] with a 52-bit key, 01101001 01101101101101101101101101101101101101101101, for our algorithm MNSKA and 4-bit key, 1000 for NSKA [5], respectively.

Table 4.1: Comparisons between Modified NSKA (MNSKA) and NSKA

Plaintext	Cipher text	
	Modified NSKA (MNSKA)	NSKA
able	$1d8ef5317a0ffccef37b1c2bfda70f4e0_{16}$	PHFT
about	$1d8ef61f7ccb435ad4fa110b3c74f9ab1_{16}$	PH^UE
above	$1d8ef61fc5f16b97e2282e64232f1a5fd_{16}$	PH^MT
everyone	$1ecbd3dd5e70f1531463301ab8803cde2_{16}$	TMTIS^NT
get	$1f6216cf654a7bcf4af81ad8abbfa6e22_{16}$	\TE
gets	$1f6216db895348e75a972b467403cbaf8_{16}$	\TEY

As seen in the table, the proposed MNSKA produces 33-bit cipher text in hexadecimal for any plaintext size of up to 16 while NSKA [5] produces a cipher text that consists of any alphanumeric value with the same size of the plaintext. Consider the plaintext "about", MNSKA encrypts the plaintext and produces the cipher text "$1d8ef61f7ccb435ad4fa110b3c74f9ab1_{16}$" Also, NSKA encrypts the same plaintext "about" and the cipher text is "*PH^MT*". We note that both algorithms are block ciphers, however, NSKA [5] encrypts one character at a time, while our proposed algorithm MNSKA encrypts plaintexts that range from size 1 to 16 as a whole and not individually.

In addition, the cipher text produced by MNSKA has no relationship to the plaintext (See Section 5) while NSKA is susceptible to frequency analysis. For example, NSKA produces the cipher text "TMTIS^NT" for the plaintext "everyone." The plaintext character 'e' occurs three times in the plaintext and the letter 'T' also presents three times in the cipher text. Since NSKA encrypts only one character at a time, one can conclude that the key can be deduced via a cipher text-only attack with the use of frequency analysis.

Two-phase Symmetric Key Encryption and Decryption

Table 4.2 presents the cipher text1 from the first-phase and cipher text2 from the second-phase. In the experiments, we use 52-bit key 00110100101101101101101101101101101101101101101101 10110110 for MNSKA in the first phase, and key1 = key2 = 642135 for the double transposition in the second phase, respectively. Each column in the table below shows the corresponding output after each phase of the encryption process. As mentioned in Chapter 3, the first phase of the encryption process uses MNSKA to convert the plaintext into its cipher text in hexadecimal form. From the table below, plaintext "*better*" is encrypted into cipher text1, "$1ddd946e417ad8fb8495dfcc97bbbc6c_{16}$" in hexadecimal form. The second phase utilizing the double transposition to transpose the cipher text1 and produces cipher text2 "$fb8d84df59cc41e67addd194bb79bcxyc6z_{16}$." Note that during the double transposition encryption process, cipher text1 is padded with three characters x, y, and z, which are later removed during the decryption.

Table 4.2: Encryption: Two-Phase Symmetric Key Algorithm

Plaintext	**Cipher text**	
	First Phase	**Second Phase**
	(MNSKA)	**(Double Transposition Cipher)**
able	$1d8ef5317ea789bb9aeb2e80609fb313d_{16}$	$bb989a2ebe807e13a78ed1f59f06b3dx31yz_{16}$
about	$1d8ef61f7ccfdae7c1a1282d90d7f24ed_{16}$	$e7adc1281a2d7cf1cf8ed1f6d709f2dxe4yz_{16}$
above	$1d8ef61fc5f60324ced010cf9c245c283_{16}$	$2430ce100dcfc5f1f68ed1f624c95c3x82yz_{16}$
better	$1ddd946e417ad8fb8495dfcc97bbbc6c_{16}$	$fb8d84df59cc41e67addd194bb79bcxyc6z_{16}$
everyone	$1ecbd3dd5e70f15318fb1868981f3aa04_{16}$	$531f1818bf685edd70cbe1d31f893a4x0ayz_{16}$

The decryption process starts with the double transposition decryption of the cipher text2, "*fb8d84df59cc41e67addd194bb79bcxyc6z*$_{16}$" to cipher text1, "*1ddd946e417ad8fb8495dfcc97bbbc6cxyz*$_{16}$". Note that the "*xyz*" at the end of the output is removed before moving onto the next phase, MNSKA Decryption process. The resulting plaintext after applying MNSKA Decryption function is "*1ddd946e417adb137be9330c3c76a0cb4xyz*$_{16}$" which is "*better*".

Security Features

In this section, we present the security features of our proposed algorithm based on the terms confusion and diffusion, and key avalanche effect.

Confusion and Diffusion

In the first-phase, MNSKA provides confusion and diffusion. Table 5.1 shows that the effect of altering one character of plaintext before encrypting with MNSKA. For the plaintext "after," MNSKA produces quotient "*1d902e6de17c9338f378*$_{16}$" and remainder "*b64dcf853a05*$_{16}$". In order to show the diffusion quality of phase one; we altered one character in the plaintext by replacing the character 'f' in plaintext "after" with a 'Q'. Thus, for the altered plaintext "aQter," MNSKA produces a quotient "*1d9023ce507d44d3a679*$_{16}$" and a remainder "*1329d5778a4f*$_{16}$".

Table 5.1: Diffusion: MNSKA

Plaintext	Cipher text	
	Quotient	Remainder
after	*1d902e6de17c9338f378*$_{16}$	*b64dcf853a05*$_{16}$
aQter	*1d9023ce507d44d3a679*$_{16}$	*1329d5778a4f*$_{16}$

As shown in Table 5.1, changing one character alters total 24 characters are altered out of 32 characters. Specifically, 13 characters out of the 20 in the quotient part and 11 out of 12 in the remainder.

In the second-phase, MNSKA followed by the double transposition, diffusion is also provided. Consider the plaintext "able", we changed the character 'b' to 'Q' giving another plaintext "aQle". After encrypting both plaintexts, the cipher text was transformed from "*aef8ec3d79c87c138b8ed1f5234fd6xy1fz*" to "*7607b82718ac89ff088ed1ec7146835x9cyz*." After comparing both cipher texts, we noticed that (1) the size of the cipher text increased from 35 to 36 and (2) the 30 out 36 characters in the cipher text was altered.

Based on our experiments, when we alter one character in a plaintext, two-phase algorithm alters over 90 % of cipher text are altered with new characters, MNSKA alters 70 – 90 % of them, and NSKA [ ] alters only one character, respectively.

Key Avalanche Criterion

According to Dawson et al [8], "A block cipher is said to satisfy the strict key avalanche effect if a one-bit change in the key causes on the average one-half of the cipher text bit positions to change."

If a one-bit in the 52-bit key of the Modified NSKA is changed, the resulting cipher text will change significantly. Table 4.2 shows the effects of altering a bit in the key. Consider the plaintext "allow", after changing one bit in the 52-bit key, the cipher text was transformed from "*87321fcf1d096b93ad91d1feec8bcdxyfbz*" to *58da216baccaab31791d1fea0dd6f3xc4yz.*" After comparing both cipher texts, 28 out of 36 (approximately 78%) characters in the cipher text were affected. We look at another plaintext "across" whose resulting cipher text was modified from "*60015323dc2174abe88fd144ec5787bxa7yz*" to "*ee7d44dc801fb3cb1d8fd1441de98exy9bz*" resulting in the alteration of 27 out of 36 (75%) cipher text characters. Since more than half of the cipher text was altered on each plaintext in Table 4.2, we can conclude that this algorithm satisfies the key avalanche effect.

In the experiments, we use a 52-bit key

Key 1 = 0011010010110110110110110110110110110110110110110110110110 in the first, and

Key 2 = 0011010010110110111110110110110110110110110110110110110110 for the experiment

Table 4.2: Key Avalanche Criterion

Plaintext	Cipher text		
	Using Key 1	**Using Key 2**	**%of altered characters**
able	aef8ec3d79c87c138b8ed1f5234fd6xy1fz	3be45b28ee43ba33bb8ed1f52f967a7x14yz	70 %
allow	87321fcf1d096b93ad91d1feec8bcdxyfbz	458da216baccaab31791d1fea0dd6f3xc4yz	78 %
herein	3f8e081b1e4b71be2baff1c931ae3axyc7z	5d045a3087dad8deb3aff1c93bd88e2x2eyz	75 %

Conclusion and Future Work

We proposed a two-phase symmetric block cipher that comprises of thirty-three to thirty-six characters as a block and 52-bit key. In the first phase, MNSKA (for Modified New Symmetric Key Algorithm) utilizes a mathematical division method and provides confusion to the encrypted cipher text and a key avalanche effect. The second phase employed the double transposition cipher to improve the diffusion.

We observed that the execution time of the encryption is roughly between one and two milliseconds for both phases. The decryption requires more time to execute compared to the encryption time. This is most likely due to a few operations that are performed in the decryption function that is not being performed in the encryption function. For instance, the decryption function removes the extra padding (x, y, z) that are added during the double transposition encryption process. It is also responsible for removing the extra random characters that are used to pad the plaintext during the MNSKA encryption process. In addition to that, the decryption function also utilizes MPIR Library to convert the decimal form of the plaintext into its binary form. Note that it is necessary to perform the additional operations in algorithm 3.1.5 in order to translate the binary form of the plaintext into its corresponding ASCII character.

For the key generation, Linear Feedback Shift Register (LFSR) was employed. This method can generate pseudo-random numbers with right-shift and exclusive-or (XOR) operations [6]. The key generation algorithm will be extended further so as to eliminate the linearity of LFSR by applying nonlinear recurrence relations and/or multiple LFSRs nonlinearly.

References

1. Mihir Bellare and Phillip Rogaway. Introduction to Modern Cryptography in UCSD CSE 207 Course Notes.2005;
2. W. Stallings. Cryptography and Newtwork Security Principles and Practice.5th ed. Prentice Hall; 2011.
3. M.J.B. Robshaw. Block Ciphers. RSA Laboratories Technical Report TR-601; 1995.
4. C. Shannon. Communication theory of secrecy systems. Bell Systems Technical Journal. 1949; 28(4): 656-715. DOI: 10.1002/j.1538-7305.1949.tb00928.x
5. Ayushi. A Symmetric Key Cryptographic Algorithm. International Journal of Computer Applications. 2010; 1(15):1–4.
6. Trappe Wade and Washington, Lawrence C. Introduction to Cryptography with Coding Theory.2nd ed. Upper Saddle River: Pearson Prentice Hall; 2006.
7. William Hart. Multiprecision Integers and Rationals (MPIR) Library. 1.2.2 ed. MPIR;2008.
8. E Dawson, H Gustafson and A N Pettitt. Strict Key Avalanche Criterion. Australasian *Journal of Combinatorics*. 1992;6:147-153.
9. K. Govinda and E. Sathiyamoorth. Multilevel Cryptography Techniques Using Graceful Codes. Journal of Global Research in Computer Science. 2011;2(7):1-5.
10. Yashaswini J. A review on Public Key Cryptography: Algorithms. International Journal of Innovative Research in Computer and Communication Engineering.2016; 4(5): 8283-8289. DOI: 10.15680/IJIRCCE.2016. 0405030.
11. Apoorva and Yogesh Kumar. Comparative Study of Differential Symmetric Key Cryptography Algorithms. International Journal of Application or Innovation in Engineering & Management. 2013;2(7):204-206.
12. Ritu Tripathi and Sanjay Agrawal. Comparative Study of Symmetric and Asymmetric Cryptography Techniques. International Journal of Advance Foundation and Research in Computer.2014; 1(6): 68-76.
13. Atul Kahate. Computer and Network security. 3rd ed. Tata McGraw-Hill Education; 2003.
14. A brief History of Cryptography.
15. N. Koblitz. A Course in Number Theory and Cryptography. Springer;1994.
16. AL. Jeeva, Dr.V.Palanisamy, K.Kanagaram. COMPARATIVE ANALYSIS OF PERFORMANCE EFFICIENCY AND SECURITY MEASURES OF SOME ENCRYPTION ALGORITHMS. International Journal of Engineering Research and Applications.2012; 2(3):3033-3037.
17. Diaa Salama Abd Elminaam, Hatem Mohamed Abdual Kader, Mohiy Mohamed Hadhoud. Evaluating the Performance of Symmetric Encryption Algorithms. International Journal of Network Security. 2010;10(3):213-219.
18. Simar Preet Singh and Raman Maini. Comparison of Data Encryption Algorithms. International Journal of Computer Science and Communication.2011;2(1):125-127.