Symbiosis

www.symbiosisonlinepublishing.com
ISSN Online: 2474-9257

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

A Secure Symmetric Key Block Cipher

Chontida Pijanchot and Eun-Joo Lee*

*Department of Computer Science, East Stroudsburg University of Pennsylvania, USA

Received: August 02, 2022; Accepted: September 10, 2022; Published: September23, 2022

***Corresponding author:** Eun-Joo Lee, Department of Computer Science, East Stroudsburg University of Pennsylvania, East Stroudsburg University of Pennsylvania, 200 Prospect Street, East Stroudsburg, PA 18301-2999, USA. E-mail: elee@esu.edu*

Abstract

Security of ciphertext is an important factor in protecting messages in transit. We propose a symmetric key block cipher to provide the enhanced security of ciphertexts and quick processing time for encryption and decryption. The proposed cipher comprises flexible block sizes and an 80-bit key. Experimental results shows that the proposed algorithm achieves confusion, diffusion, and key avalanche effect which are desirable characteristics of secure cryptography. As a result of the experiment, the average change rate of the ciphertext is 93.99% when 1 bit is changed in the plaintext. In addition, when 1 bit is changed in the 80-bit key, the ciphertext is changed to an average of 93.99% and the maximum to 100%.

Key words: Cryptography; Symmetric key cipher; Block cipher; Confusion; Diffusion;Avalanche effect.

Introduction

It is important to be careful when using computer systems because of security threats. Information in electronic format may be viewed or changed while stored or communicated over a computer network. Therefore, the implementation of a reliable and efficient computer security system is essential [1-4].

Cryptography is the process of transforming plain data into a format that the general public cannot understand. In cryptography, a cipher is an algorithm for performing encryption or decryption. Symmetric key cipher refers to an encryption/decryption method in which the sender and receiver share the same key. Block ciphers encrypt data in groups or blocks, whereas stream ciphers encrypt a digital stream one bit or one byte at a time [3, 5]. It is generally considered that stream ciphers take less encryption/decryption processing time than block ciphers do. However, it is also considered that block ciphers could be more secure than stream ciphers because of their properties of confusion and diffusion [6-10].

Ayushi developed an algorithm New Symmetric Key Algorithm (NSKA) using a simple mathematical method [11]. NSKA is simple in nature and works very smoothly for small amounts of data. However, NSKA is a stream cipher and is weaken to a brute-force attack. Lee, Omo-Ekpadi, and Kimm [12] proposed Modified Symmetric Key Algorithm (MNSKA) that compensates for the weakness of NSKA[11]. MNSKA generates a 33-bit ciphertext by encrypting 16-bit characters as a block with a 52-bit key, and provides confusion to the encrypted ciphertext and a key avalanche effect.

As a continuous work of MNSKA [12], we develop an algorithm which requires less processing time than MNSKA and improves the level of security. In this paper, we propose a secure symmetric key block cipher (SSKBC) that uses a flexible block size with an 80-bit key. The proposed algorithm compensates MNSKA by using a flexible block size rather than a fixed block size (16-bit), so the encryption and decryption execution time is shortened. The algorithm also enforces an 80-bit key applied by a non-linear or linear pseudo-random number generator for added security. Experimental results show that the proposed SSKBC improves security by increasing the diffusion rate by about 20% compared to the previous algorithm, MNSKA. In addition, the processing time of SSKBC reduces 2 to 4 times using an appropriate block size of plaintext compared to MNSKA.

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

Modified New Symmetric Key Algorithm

In this section, we first review the general framework of Modified New Symmetric Key Algorithm (MNSKA) [12]. Algorithm 3.1 is a symmetric block cipher based on a simple mathematical method that is related to the New Symmetric Key Algorithm by Ayushi[11].

Algorithm 3.1 Encryption: Modified New Symmetric Key Algorithm (MNSKA)

1. Generate a 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 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. Ciphertext C = Q + R

This algorithm proposed two-step encryption in block size 16-bit character as a block with a 52-bit key generated from Linear Feedback Shift Register (LFSR) [8]. It produces a 33-bit ciphertext in hexadecimal form for any plaintext size up to 16 letters.

MNSKA uses a fixed block size (16 bits), so if the plain text size is greater than 16, more blocks are required. For example, if the plaintext size is 32, MNSKA needs 2 of the 16-bit blocks for encryption and decryption. Therefore, it takes about twice the time to process 1 block. Also, if the size of the plaintext is 8 smaller than 16, processing time for a 16-bit block is required.

A Secure Symmetric Key Block Cipher (SSKBC)

We now introduce a secure symmetric key block cipher (SSKBC) based on a mathematical partitioning method[11]. A continuation of previous algorithm MNSKA[12], we consider a flexible block size based on the input plaintext size. A secure symmetric key block cipher (SSKBC) determines a block size based on the input message to reduce execution time and improve security using 80-bit keystreams that generate more secure ciphertext. Rivest Cipher 4 (RC4) [5] or the Linear Feedback Shift Register (LFSR) [8] is utilized to generate an 80-bit key K.

Algorithm 3.2. Encryption: Secure Symmetric Key Block Cipher (SSKBC)

1. Generate an 80-bit key K

2. Determine the block size, B, from 8 to 32 based on the size of plaintext P

3. If the size of plaintext P is less than block size B, then pad plaintext P with additional characters up to the maximum size of block size B

4. Convert plaintext P to ASCII and then to decimal form

5. Convert K from binary to decimal form

6. Compute Quotient Q and remainder R by P/K

7. Convert Q and R to hexadecimal

8. Ciphertext C = Q + R

Algorithm 3.1 uses different block cipher sizes with block sizes of 8, 16 and 32 depending on the length of the input message. Encrypts 8, 16, and 32 characters (plaintext P) into blocks and produces 17, 35, and 65-bit ciphertext in hexadecimal, respectively. The ciphertext consists of a quotient and a remainder, respectively, by dividing the plaintext and key-values. For the decryption, the following algorithm 3.2 shows the decryption process of SSKBC.

Algorithm 3.3. Decryption: Secure Symmetric Key Block Cipher (SSKBC)

1. Extract Quotient Q and Remainder R from the ciphertext C

2. Convert quotient Q and remainder R from Hexadecimal to Decimal form

3. Convert key K from binary to decimal form

4. Compute plaintext P multiplying by key K and adding the remainder R to it, P = Q * K + R

5. Convert plaintext P from decimal to binary form and then to ASCII

6. Remove the padding

In the encryption process, after obtaining the ciphertext in the step 8, the ciphertext is rearranged to make diffusion of the ciphertext by applying a double transposition. The double transposition utilizes two different keys for each transposition. The decryption process begins with the double transposition to return the rearranged ciphertext back to the ciphertext generated by Algorithm 3.2.1. It then applies SSKBC decryption algorithm to recover the original plaintext.

Experimental Results

In this section, numerical experiments of the proposed encryption algorithm are presented. Two key generation algorithms, non-linear (RC4 [1]) and linear (LFSR [3]) pseudo-random number generators, are utilized to generate 80-bit keys. RC4 uses the initial character "*computersc*" and LFSR uses the initial state of "*01101000010*".

All the tables and figures in this section contain numeric results and use 80-bit keys using RC4,00010100010111010110101001 11100001001010101100100010011101010100000000010101 10000.

Key Generations

The key generation algorithms are developed based on the Rivest Cipher 4 (RC4) [5] and the Linear Feedback Shift Register (LFSR) [8]. RC4 is a non-linear and LFSR is a linear algorithm used to produce an 80-bit key with 10 initial characters and 10-bit initial vector, respectively. The period of the RC4 algorithm could be up to 10^{100} whereas the LFSR will have 2^n-1different states [3,4 and 5].

To estimate and compare the average key generation times of the RC4 and LFSR algorithms, we generate 10 keys containing 80 bits of each key. Figure 4.1 shows that 10-key generation times using RC4 ranged from a minimum of 3.4 milliseconds to a maximum of 8.7 milliseconds per key generation. The generation times of LFSR range from a minimum of 14 milliseconds to a maximum of 25.5 milliseconds each. The average 80-bit key generation times using RC4 and LFSR are 0.503 milliseconds and 1.852 milliseconds, respectively.

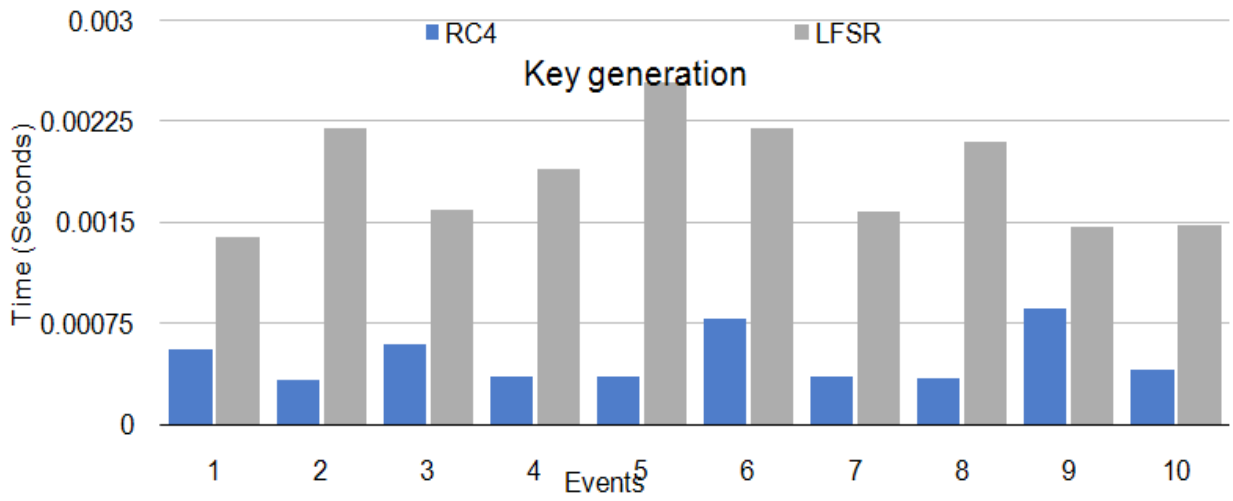

Figure 4.1 : Comparisons of Key Generation Time between RC4 and LFSR

Secure Symmetric Key Block Cipher (SSKBC)

SSKBC can utilize various block sizes to generate ciphertext. Our experiments use 8, 16, and 32-bit block sizes, and SSKBC produces 17, 35, and 65-bit hexadecimal ciphertexts with the block sizes, respectively. Table 4.1 shows the ciphertexts generated using the different block sizes. For a plaintext size of 8, SSKBC generates 17, 35, and 65-bit ciphertext using 8, 16, and 32-bit block sizes, respectively. For a plaintext size of 16, SSKBC generates 34 (=2*17), 35, and 65-bit ciphertexts using 8, 16, and 32-bit block sizes, respectively. Since the plaintext size is 16 and the block size is 8, SSKBC has to process it twice to get the ciphertext. If the size of the plaintext is 32, 68 (=4*17), 70 (=2*35), and 65-bit ciphertexts are generated for the same reason as above.

If the block size is smaller than the plaintext size, the encryption process will have to do more than one time to encrypt the plaintext and it will take longer to process. Figure 4.2 shows the encryption processing times for plaintext size = 16 with block sizes 8, 16, and 32 using RC4. As shown in the figure, encryption processing time of block size = 16 takes shortest time to encryption process when plaintext size = 16. In addition, using an appropriate block size of plaintext, encryption processing time can be reduced by 2 to 4 times.

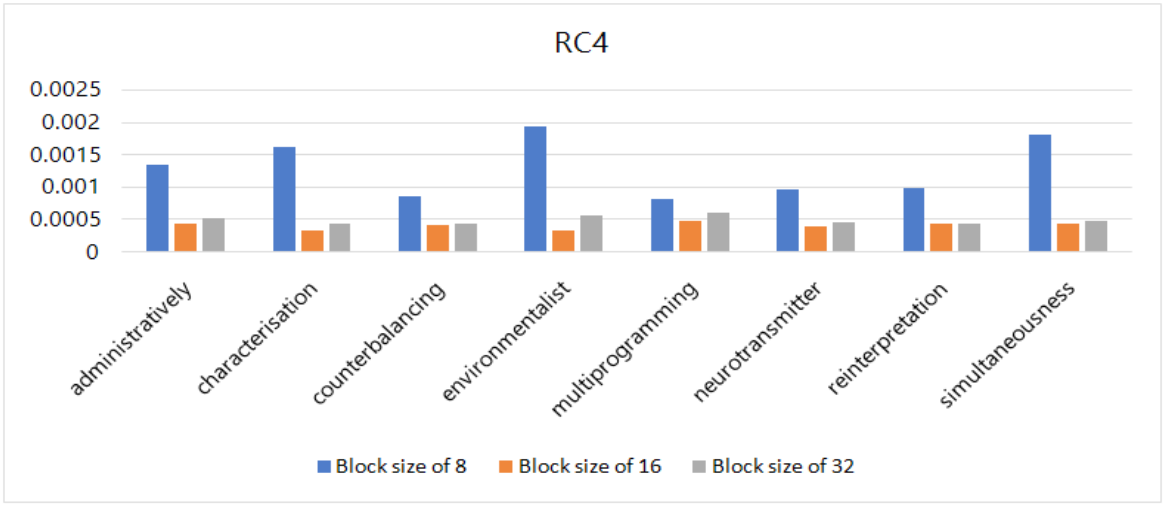

Figure 4.2 : Processing Time Compari'sons of various block sizes using RC4

Table 4.1 Comparisons of ciphertexts using different block sizes using the key (RC4)			
Plain Text	**Cipher Text**		
	Block size = 8	**Block size = 16**	**Block size = 32**
American	6016E75626963616D	2D80F79Y951AFF45EY 35BFE45AEACD7B1A0	37E50E201CE20752062 E78DB6D5A3FA43C7DE 87B0591A011F909 E176021C6E6EF
Birthday	6.01797E+16	6FF1897L9E308CLCCL F581CFBDD97BDB157	4084E98C17F24A2D984 CFA86417AF26EA4BA EC0F3512EFE6B75 BF62E53E0C5AT0
Administratively	603746D696E796164 70C79647976656261	8CB5407LE4A64B17AL 119424879F8967126	55B66BAC9E1467EB9 F21B37296C0AF68C C94FB28E5D879DBE0 D834441FE2E07DD
Autointoxication	6046F647F697E6175 70F6E636174696869	9426DDCNDEC729N4D N91B35ED13A9B6416C	A03B5E260F7F77A9 EC205D00FFA0DE1 1A0FEB764849C1EB C68273C91B221A06S0
Cryptography prior to the modern ages	607727970746F63726096F 78697062717070D6F7F64 6865627460573627E61676465	43B3F36S43FB4A511S211F083D ED1FCB193B6207ACS1FFDB D334S2470E4434E3B6B1C6	06E47E8531BA59B3B 87F0BD53D6FFA1 67E4EEAC234612F 11CAE1991F04D52AA3B
was effectively synonymous with encr	70563637566666761703796 6756C797469605737E69 6D7F6E6F703726478656E6769	DA31548KB9243D3E5 K2559FEB9E800461 5D55D5724KABC45CK B8K5391648783C2BC15A	E7C95DB1B967C42B 2C2C54952113E42E8F 2916B3D57CC404B943 A5BF7628278K7

Security Features

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

The property of confusion makes it difficult to find the key from the ciphertext and if a single bit in a key is changed, the calculation of most or all of the bits in the ciphertext will be affected [6, 9, 10]. Tables 4.2 and 4.3 show that when 1 bit is changed in the 80-bit key of SSKBC, the resulting ciphertext is changed from 91% to 100% using RC4 keystream. In the table 4.2, key1 and key2 are

Key1 = 00010100010111010110101001111000010010101011001000100111010101000000001010110000, and

Key 2 = 10010100010111010110101001111000010010100011001000100111010101000000001010110000.

Table 4.2 Key Avalanche Criterion using 16-bit block size and RC4 keystream			
Plaintext	**Ciphertext using**	**Ciphertext using**	**Percentage difference between two ciphertexts**
	Key 1	**Key 2**	
Sunny	2C54AE5F187436521F44D6DACBF93865138	AB9CDF8E71F59666FA1BA0D02ED5551C00D	94.29%
Metropolitan	BB0F2A51A520591B41B5E29CC981E6DF1AC	AAF9A2880C66E666E9E29F715E9374EB6FE	100%
Rainy	85FF536564AB4C3A1544C949130A68FE10E	2845398D3B5C9777FC62DC9FC9691553E08	97.15%
Weather	4A21C5CD880EF30E7D54C3A96D7905C2177	E291DD9D6D6A28884E07BCB7AD849484ECC	91.43%
Computer	FC9ACDA4F581864C5424870197E83A1117C	2E05999F1624BBBB2D8BEF5DFD1155228F5	100%

Table 4.3 shows the experimental results of changing the 1-bit keystream generated by LFSR, and the resulting ciphertext is changed from 85% to 100%. In the table, key1 and key2 are

Key1 = 10101000010101000010101000010101000010101000010101000010101000010101000001010, and

Key 2 = 00101000010101000010101000010101000010101000010101000010101000010101000001010.

Table 4.3 Key Avalanche Criterion using 16-bit block size and LFSR keystream

Plaintext	Cipher text using	Cipher text using	Percentage difference between two ciphertexts
	Key 1	Key 2	
Sunny	8B4BCED72C027DDD72D9CECFDC24FAF9BA7	1821EF48B5079666FB72D510454C7208C39	97.14%
Metropolitan	464BBAF453A58AAA537C9360699DE9E836E	B62A03AC17FE9D8DBFCC9A4CD186D26DF2B	91.43%
Rainy	8B4A481C079EAAAAF06EC00BDE1A0B198A7	BEDA6C33F1205AAA8BDB385531187266D3B	85.71%
Weather	2FCC877B4D5A2222C130E9024AC33AE4B1A	63C45C5D16124D8D5A5308FA3058D2971CE	97.14%
Computer	7A0C8567638322226895F86787410A0C35E	B26A34B1A5C19999AFA754C4768DA23CC1A	97.14%

Tables 4.2 and 4.3 show that a change of 1 bit in an 80-bit key in SSKBC significantly changes the resulting ciphertext. This allows this algorithm to satisfy the key avalanche effect.

According to Stallings [5], "Diffusion means that if we change a single bit of the plaintext, then about half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then about half of the plaintext bits should change." Diffusion based on our experiments is also presented in the proposed algorithm. Table 4.4 and 4.5 show that the effect of changing one character in plaintext encrypting with SSKBC using RC4 and LFSR keystreams, respectively.

In Table 4.4, the percentage difference between the two ciphertexts is 75.76% to 82.09% when using the RC4 keystream. Table 4.5 shows that the percentage differences are higher when using LFSR keystreams than when using RC4. The percentage ranges from 88.58% to 100%.

Table 4.4 Diffusion: SSKBC using 16-bit block size and RC4 keystream

Plaintext 1	Ciphertext (Plaintext 1)	Plaintext 2 – altering one character in Plaintext 1	Ciphertext (Plaintext 2)	Percentage difference between two ciphertexts
Academic	E9E3475A2D36DFAA9 AD298AAC172CD1411F	xcademic	13F3AA81C7069E3E1BCD9E6377FC2157305	75.76%
Political	4E833FDFC9D76D00D F559ABD344569A9101	aoliticals	EB6D5F0525DF313146268FE278E4516E581	82.09%
Metropolitan	BB0F2A51A520591B4 1B5E29CC981E6DF1AC	aetropolitan	EAA944331C9FD1C138A2479B38359126DF3	82.09%
Physical	A17204881A7CED6DC 8329994A2B46F4215E	ahysical	444324F48665D525249176EEA683A1C8964	76.12%

Table 4.5 Diffusion: SSKBC using 16-bit block size and LFSR keystream

Plaintext 1	Ciphertext (Plaintext 1)	Plaintext 2 – altering one character in Plaintext 1	Ciphertext (Plaintext 2)	Percentage difference between two ciphertexts
academic	6B71E53387395111E 4A53A701C803A46ED3	xcademic	E0E11BB627EF5AAA30 D681BE4686BB31727	88.58%
political	EB14E8762DD162228 5C0F5CDE7C4AAF78B4	aoliticals	A4AE6D56D4111777053 507A65E1289C8C78	91.43%
metropolitan	464BBAF453A58AAA5 37C9360699DE9E836E	aetropolitan	D0F415AB289AFFFFA8 ADED7DEB443A99E0F	100%

Comparison of Secure Symmetric Key Block Cipher (SSKBC) and Two-Phase Symmetric Key Block Cipher (TSKBC)

In this section, the proposed Secure Symmetric Key Block Cipher (SSKBC) algorithm and the Two-Phase Symmetric Key Block Cipher (TSKBC, [12]) algorithm are compared in terms of diffusion.

In Table 4.6 and 4.7, SSKBC uses RC4 and LFSR to generate 80-bit keystreams, while TSKBC uses LFSR to generate a 52-bit keystream. For SSKBC, the 80-bit RC4 keystream is 00010100010111010110101001111000010010101011001000100111010101000000001010110000, and LFSR keystream is 00010100010111010110101001111000010010101011001000100111010101000000001010110000. For TSKBC, the 52-bit key stream is 0011010010110110110110110110110110110110110110110110110110.

Table 4.5 Diffusion: SSKBC using 16-bit block size and LFSR keystream

Plaintext 1	Ciphertext (Plaintext 1)	Plaintext 2 – altering one character in Plaintext 1	Ciphertext (Plaintext 2)	Percentage difference between two ciphertexts
academic	6B71E53387395111E 4A53A701C803A46ED3	xcademic	E0E11BB627EF5AAA30 D681BE4686BB31727	88.58%
political	EB14E8762DD162228 5C0F5CDE7C4AAF78B4	aoliticals	A4AE6D56D41117770 53507A65E1289C8C78	91.43%
metropolitan	464BBAF453A58AAA 537C9360699DE9E836E	aetropolitan	D0F415AB289AFFFFA 8ADED7DEB443A99E0F	100%

Table 4.6 shows that for a plaintext size of up to 16 with a block size of 16, SSKBC generates 34-bit hexadecimal ciphertext while TSKBC generates 33-bit ciphertext, respectively. To compare the diffusion effect of the two algorithms, one character of the plaintext is changed and the ciphertext of the two plaintexts is displayed in the table.

Table 4.7 shows the percentage difference between the two ciphertexts using TSKBC[12] and the proposed algorithm SSKBC. For the proposed SSKBC algorithm, considering the plaintext "immediate" and "xmmediate" and both the ciphertext, the ciphertext character change rate is 94.29% using RC4 and 91.43% using the LFSR keystream, respectively. However, using LFSR for TSKBC algorithm gives a 75% rate of change. Experimental results show that this algorithm, SSKBC, improves security by increasing the spread rate by about 20% compared to the previous algorithm, TSKBC.

Table 4.7 Comparison of percentage differences between TSKBC & SSKBC
Table 4.7 Diffusion: Comparison of SSKBC and TSKBC using 16-bit block size

Plaintext 1	Plaintext 2 – altering one character in Plaintext 1	Percentage difference between two ciphertexts		
		TSKBC (LFSR)	SSKBC (RC4)	SSKBC (LFSR)
across	xcross	75%	88.57%	97.14%
after	xfter	72%	82.86%	97.14%
hereafter	xereafter	73%	85.71%	85.71%
indicated	xndicated	81%	94.29%	94.26%
immediate	xmmediate	75%	94.29%	94.26%

Conclusion and Future work

We proposed a secure symmetric key block cipher (SSKBC), an updated version of Modified New Symmetric Key Algorithm (MNSKA[12]). SSKBC uses multiple block sizes based on text size to reduce processing time and improve security. Three block sizes were used in this paper: 8, 16, and 32. Applying an appropriate block cipher size in plaintext makes the encryption and decryption process effective. For key generation, use 2 algorithms to produce an 80-bit key which is RC4 and LFSR.

Experimental results show that SSKBC reduces 2 to 4 times of encryption/decryption processing time using an appropriate block size of plaintext compared to MNSKA. Also, the average change rate of the ciphertext is 93.99% when 1 bit is changed in the plaintext. In addition, when 1 bit is changed in the 80-bit key, the ciphertext is changed to an average of 93.99% and the maximum to 100%. SSKBC improves security by increasing the diffusion rate by about 20% compared to the previous algorithm, MNSKA.

For future work to improve security features if plaintexts have the same length as block cipher size, the algorithm might force plaintexts to encrypt with the larger block cipher size. Since plaintext that has the same length as block cipher size would generate the same cipher. For the time for key generation, we observed that the execution of LFSR key generation is roughly between one to three milliseconds while RC4 takes less than one millisecond. This is most likely due to the operations that

are performed between different registers. The time processing could be faster if we operate under the same register.

References

1. R. Smith, Internet Cryptography. Pearson Education. 2004;
2. R. Smith, Internet Cryptography: Evaluating Security Techniques. Addison-Wesley Professional.1997;
3. B. Forouzan, Intorduction to cryptography and network security. McGraw-Hill Higher Education.2008;
4. O. C. a. i. affiliates, "https://docs.oracle.com/cd/E19424-01/820-4811/aakfw/index.html," Oracle Corporation, 2010. [Online]. Available: https://docs.oracle.com/cd/E19424-01/820-4811/aakfw/index.html.
5. W. Stallings. Cryptography and Networking Security Principles and Practices. Prentice Hall. 2011;
6. C. Shannon, "Communication Theory of Secrecy Systems," Bell System Technical Journal. 1949;28(4): 656-715.
7. P. R. C. S. P. &. T. A. Ellingsen, "An extension of the Avalanche criterion in the context of C-differentials," in Proceedings of the 18th International Conference on Security and Cryptography, 2021.
8. Trappe Wade Lawrence C. Washington. Introduction to Cryptography with Coding Theory.Upper Saddle River: Pearson Prentice Hall. 2006;564-570.
9. L. Battern, Public Key Cryptography: Applications and Attacks. Wiley-IEEE Press. 2013.1-224;
10. Rohith V. Sharma AK. Cryptography: Avalnache Effect of AES and RSA. International Journal of Scientific and Research Publications.2020. 10(4); 119-125.doi: 10.29322/IJSRP.10.04.2020.p10013
11. A. Ayushi, "A Symmetric Key Cryptographic Algorithm," International Journal of Computer Applications.2010.1(15); 1-4, doi: 10.5120/331-502
12. Kimm H, Joo Lee E, Bose OE (2019) A Two-Phase Symmetric Key Block Cipher. J Comp Sci Appl Inform Technol. 4(1): 1-5.
13. S. Kullback, Information Theory and Entropy. Model Based Inference in the Life Sciences: A Primer on Evidence. , New York, Springer, 2008;51-82.doi: 10.1007/978-0-387-74075-1_3