International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 3 (2019) pp. 604-608

A New Permutation Based Randomized Neighborhood-Xor Modulo Algorithm

[1]Dr. S. Arul jothi*, [2]Dr. S. Pushparaj

[1]Assistant Professor, Department of Computer Science, Fatima College, Madurai, Tamil Nadu-625018, India.

[2]Associate Professor & Head, Department of Econometrics, School of Economics, Madurai Kamaraj University, Madurai, Tamil Nadu-625 020, India.

Abstract

Cryptographic algorithm of different schemes is at various stage of development with different cryptanalytic characteristics. This paves the way for booming and competent security algorithms. Encryption schemes are based on block or stream ciphers. Comparing different block cipher schemes with respect to their performance are not common in the literature. Some of the block cipher algorithms are used commonly for their known performance capabilities and determent to cryptographic attacks. This paper attempts to compare the newly developed Neighborhood-Xor Modulo algorithm within the class of block cipher algorithms viz., DES, AES, RC6 and Blowfish.This algorithm also undergoes FIPS 140-2 statistical test for randomness.

Keywords Block Cipher, Symmetric, FIPS 140-2, Throughput, Execution Time.

INTRODUCTION

As the electronic network has become universal and that network is vulnerable to hacker and virus attacks, electronic fraudulence and wiretap has become a common place, securing the computer network speculate significance. Cryptography has been emerging regularly to meet the differential demands of security concerns as discussed in [1]. The cryptography has to forge new methods continuously to serve this purpose, as the security is breached with ever increasing capacity of hackers that grows with their computing power.

The conventional cryptographic algorithms entrust on the combination of substitution and transformation strategy to safeguard information from attack. Countering the cryptographic attack requires an efficient strategy with ever increasing capabilities of computers. To enable the potency of the current algorithms various strategies are adopted in practice. Common crypt attacks are based on the uniformity of logical frame changing the logical framework from the traditional logic of cryptography can grasp the attempts of the hackers because the innovation in basic logical frame would make any extent of attack trivial.

The prevailing study introduces such change in the scheme of encryption. Here in this algorithm the key is nothing but the random positions of the given n × n matrix. Therefore it is defiant to all of the input blocks.

RELATED STUDIES

Cryptology constitutes two interdependent fields: cryptography and cryptanalysis. Cryptography is the science and art of keeping information safe from unintended users, of encrypting it as explained in [2]. Cryptanalysis is the art and science of breaking encoded data, that is, to estimate and explore design countenance that may lead to the exploration of some piece of secret information.

A random number generator (RNG) as defined in [3] is a mathematical construct that is designed to generate a random set of numbers that should not display any prominent patterns in their emergence or propagation, hence the word random. The generation of random numbers is essential to cryptography. They are of importance in the construction of encryption keys and other cryptographic algorithm parameters. In practice statistical testing is employed to gather evidence that a generator produces numbers that appear to be random. Few resources are readily available to researchers in academia and industry who wish to analyze their newly developed RNG. To address this problem, NIST discussed in [4] has developed new metrics that may be employed to investigate the randomness of cryptographic RNGs.

FIPS 140-2 is a U.S. government computer security standard used for validating the effectiveness of cryptographic hardware, that is hardware or software that encrypts and decrypts data or performs other cryptographic operations as defined in [5]. FIPS 140-2 specifies security requirements that are to be satisfied by a cryptographic module used within a security system protecting information within computer systems. A crypto module is any combination of hardware, firmware, and software that implements such cryptographic functions as encryption, hashing, key management, or message authentication.

Keys play an important role in encryption and decryption. If weak key is used in algorithm then every one may decrypt the data. Strength of Symmetric key encryption depends on the size of the key used as described in [6]. For the same algorithm, encryption using longer key is harder to break than the one done using smaller key. There are many examples of strong and weak keys of cryptography algorithms like DES, RC6, Blowfish, and AES. DES uses one 64-bit key, while AES uses various (128,192,256) bits keys. Blowfish uses various (32-448); default 128-bit while RC6 uses various

(128,192,256) bits keys. Our algorithm uses (64,128, 256, 512, 1024) bits key and this has been discussed in next section.

PROPOSED ALGORITHM

This new proposed algorithm is a block cipher that divides data into blocks of equal length and then encrypts each block using a randomized key as discussed in [7]. In the proposed algorithm, block size of the plain text and key size is same and flexible. Selection of the key is purely random based and it uses the same key at both ends. This scheme requires less computation as it uses bitwise operations and highly sensitive to small changes in the secret key used in the algorithm.

This paper introduces a permutation generation mechanism based on a shared secret key. The generated permutation, are used as encryption keys in a block ciphering cryptosystem. The technique makes use of position permutation and involves three different phases in the encryption process. In the first phase the plaintext is divided into blocks of equal size. The second phase is the key generation phase, where the bit positions in the plain text block are used to build a key.

The technique proposed is a unique one from the others in a way that the key is generated, the random positions of the plaintext. Further permutation is applied based on the dynamic key to strengthen the encryption. In the third phase the blocks are encrypted. The technique proposed is a unique one from the others in a way that the key is generated with valid information about the values used in the encryption process.

In most of the encryption processes the key is generated first and then do the encryption process. The technique generates a relation between the encryption process and the key. The proposed algorithms uses larger key, selected using random generators, for different permutation operations. Since larger keys are used, the security level offered is also high. Further, the amount of redundant information available in the encrypted file is kept as low as possible and thereby it provides high security against attackers.

Pseudo Code of Proposed Encryption algorithm

A. key generation

Generate a pair of values (i,j) randomly, 1<=i<=m, 1<=j<=m, such that no pair (i,j) is repeated more than once.

Let π denote the collection of pairs (i,j) stored as per the order of generation.

B. Encryption

Let P be a plaintext of size n.

Let m be the size of the block.

Let $a = n \bmod m^2$.

If $a = 0$ set $k = n/m^2$

else set $k = n/m^2 + 1$.

The plaintext P is divided in to k blocks, say $P_1,P_2,---,P_k$, such that the first m entries of P form the first row of P_1, second m elements the second row of P_1 and so on.

Similarly we define $P_2,P_3,---,P_k$.

If $a \neq 0$ then we add the require number of zeros at the end of the plaintext, so that the last block P_k is of order $m \times m$.

For $l = 1,\ldots,k$.

Let a_{ij} be the (i,j)th element of P_1.

Change a_{ij} as per the generation order of the pair(i,j) as follows,

We calculate the modified $(n \times n)$ binary matrix a_{ij}.

$$a_{ij} = a_{i-1j-1} \oplus a_{i-1j} \oplus a_{i-1j+1} \oplus a_{ij-1} \oplus a_{ij} \oplus a_{ij+1} \oplus a_{i+1j-1} \oplus a_{i+1j} \oplus a_{i+1j+1}$$

Let C_1 be the $m \times m$ matrix with the modified a_{ij}.

C_1 is the ciphertext corresponding to P_1.

C. Decryption

For $l = 1,---,k$.

Let a_{ij} be the (ij)th element of C_1.

Change a_{ij} as per the reverse order of generation of the pair (i,j).

The $(a_{ij}) = P_1$, the original plaintext block.

The plaintext $P = P_1,P_2,---,P_k$.

SECURITY ANALYSIS

The proposed crypt algorithm is tested for its encryption quality. For the experiment, Microsoft Windows XP Professional Version 2002 Service Pack 3 on, Intel(R) Core(TM) Duo CPU, 1.83GHz to 0.99 GB of RAM is used, in which performance data is collected.

Test is done on the following three bmp images of size 512×512 as shown in Fig. 1

Barbara

Peppers

Clown

Figure 1: Test images

Simulation program is implemented using NetBeans IDE. The results of the encryption quality were analyzed using Matlab 7.0.

A. Statistical random number generator tests

Cryptographic modules that implement a random or pseudorandom number generator shall incorporate the capability to perform statistical tests for randomness. The tests specified below are recommended. However, alternative tests which provide equivalent or superior randomness checking may be substituted. A single bit stream of 20,000 consecutive bits of output from the generator is subjected to each of the following tests. FIPS 140-2 publication for cryptographic modules specifies four statistical tests for randomness. Instead of making the user select appropriate significance levels for these tests, explicit bounds are provided that the computed value of a statistic must satisfy. If any of the tests fail, then the generator fails the FIPS 140-2 statistical test for randomness as given in [8]. The distribution functions are derived and significance levels were calculated for each of the four statistical tests. The observed statistical values of our algorithm are also shown with the calculated theoretical values.

1) The Monobit test

The number of ones is counted in a bit stream of 20,000 bits. Considering a bit stream of 20,000 bits in which all bit occurrences can be considered as Bernoulli trials with success probability of ½. Let x denotes the number of ones occurred in n bits, then the number of distinct patterns are ${}_nC_x = \frac{n!}{[(n-x)!x!]}$. The probability distribution function f(x) describes the probability of x number of ones in n bits. In Table 3, we calculate the number of ones in the 20,000 bit stream generated by our algorithm with three different keys as in [9].

Statistic x: denotes the number of ones in the bit stream.

Acceptance Region: $9{,}654 < x < 10{,}346$.

Table 1: Monobit Test for Three images with Three Different Keys

IMAGES	KEY 1	KEY 2	KEY 3
Barbara	9981	10083	10047
Peppers	10004	9973	10074
Clown	10033	10095	9969

Thus all the three images are in the accepted region.

2) The Poker test

Divide the 20,000 bit stream into 5,000 contiguous 4 bit segments. Count and store the number of occurrences of each of the 16 possible 4 bit values as discussed in [10]. Denote f(i) as the number of each 4 bit value i where $0 < i < 15$. Then, we evaluate the following:

$$x = \left(\frac{16}{5000}\right) \times \sum_{i=0}^{15} g_i^2 - 5000 \qquad (5)$$

Statistic x:

$$x = \left(\frac{16}{5000}\right) \times \sum_{i=0}^{15} g_i^2 - 5000 \qquad (6)$$

Acceptance Region: $1.03 < x < 57.4$.

Table 2: Poker Test for Three Images with Three Different Keys

IMAGES	KEY 1	KEY 2	KEY 3
Barbara	11.4624	26.9311	21.1264
Peppers	24.1152	29.0496	12.7615
Clown	11.3663	14.6559	23.9423

Thus all the three 20,000 bit stream generated from three sample images by our algorithm with three different keys are in the accepted region.

3) The Runs Test

A run is defined as a maximal sequence of consecutive bits of either all ones or all zero, which is part of the 20,000 bit sample stream. Test is individually performed for each length of 1, 2, 3, 4, 5 and 6 of either one or zero. The test is passed if the number of runs that occur (of lengths 1 through 6) is each within the corresponding interval specified below in Table 5. This must hold for both the zeros and ones; that is, all 12 counts must lie in the specified interval [11].

For the purpose of this test, runs of greater than 6 are considered to be of length 6. Three sample images with three different keys are tested for run test and the results are given in Table 6 given below. The test is passed if the number of runs that occur is each within the specified interval as stated in Table below.

Statistic x: denotes the number of runs of each length appear in the bit stream.

Table 3: Acceptance Regions for Run Test

Length of runs	Acceptance region
1	$2{,}315 \le x \le 2{,}685$
2	$1{,}1114 \le x \le 1{,}386$
3	$527 \le x \le 723$
4	$240 \le x \le 384$
5	$103 \le x \le 209$
≥6	$103 \le x \le 209$

Table 4: Runs Test for Three Images for two Different Keys

Length of runs	Barbara		Peppers		Clown	
	Key1	Key 2	Key1	Key2	Key1	Key 2
1	2546	2494	2512	2541	2482	2560
2	1288	1293	1242	1311	1251	1259
3	648	605	618	637	628	619
4	317	330	309	320	300	311
5	165	152	164	163	158	150
≥6	165	168	159	166	135	152

4) Long Run Test

The length of the longest run is examined for the bit stream of 20,000 bits. Statistic x: the length of the longest run in bit stream of 20,000 bit(both of one and zero). Acceptance Region: $x < 26$. A long run is a run that formed by '0' or '1' with length 34 or more[12]. The test is passed if there are no long runs on the 20,000 consecutive bits.

In Table 7, we calculate the longest run in the 20,000 bit stream generated by our algorithm with three different keys. Three sample images with three different keys are tested for long run test and the results are in the accepted region.

Table 5: Long Run Test for Three Images for two Different Keys

IMAGES	KEY 1		KEY 2		KEY 3	
	1's	0's	1's	0's	1's	0's
Barbara	15	17	14	12	12	13
Peppers	14	12	13	15	13	17
Clown	15	13	13	16	12	12

B. Execution Time and Throughput

To investigate the relative performance of proposed algorithm the encryption time and throughput analysis is done. An important tool to evaluate the efficiency of algorithms is measuring the amount of time required to encrypt a file as given in Ozturk et al., [13] The encryption time is the time that an encryption algorithm takes to produce a cipher text from a plaintext. It indicates the speed of encryption. The results of this test are shown in Table 6.

Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total plaintext in bytes encrypted divided by the encryption time. The throughput of the encryption scheme is calculated as in equation below.

$$\text{Throughput} = \frac{T_p(\text{MegaBytes})}{E_t(\text{Second})} \qquad (1)$$

Where, Tp: total plain text (megabytes)
Et:encryption time(second)

Table 6: Comparative Execution Time of various Encryption Algorithms for Different Files with RNXA

File Type & Size	DES	AES	RC6	BlowFish	RNXA
Txt, 4kb	2.876	2.412	1.934	1.567	1.067
Xls, 44kb	3.879	3.423	2.987	2.134	1.074
Doc, 551kb	4.786	4.287	3.837	3.256	2.112
Pdf, 1.20mb	6.482	5.936	5.489	4.982	3.264
Jpeg, 1.69 mb	8.234	7.923	6.734	5.567	4.156
Bmp, 2.25mb	9.446	8.345	8.132	6.932	5.784
Mp3, 7.42 mb	11.99	10.92	9.645	8.234	5.309
Vob, 56.6 mb	72.55	70.34	66.12	60.011	54.523
Average Time	15.03	14.19	13.11	11.5853	9.6611
Througput (mega bytes/sec)	4.630	4.902	5.309	6.00798	7.2046

The table clearly shows that for all types of files the time taken for encryption in the new algorithm is the least. Similarly the throughput is the highest for the new algorithm compared to the commonly used encryption algorithms viz., DES, AES, RC6, Blowfish.

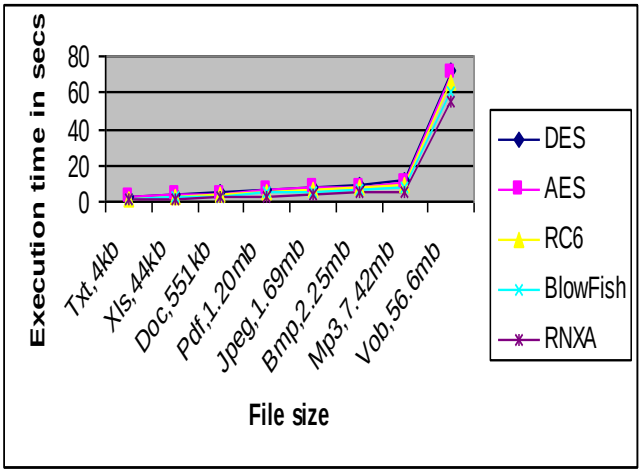

Figure 2: Encryption time for different files

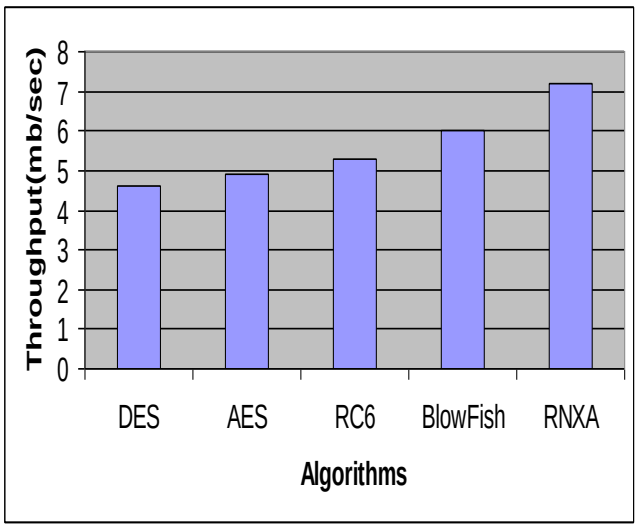

Figure 3: Throughput for different files

The average comparative execution time of select encryption algorithms for different file formats are shown in the bar chart given above.

CONCLUSION

Proposed algorithm proved to outperform popular encryption schemes in terms of stability and execution time. This result indicates that the new algorithm had the least encryption time and highest throughput when compared to selected symmetric encryption algorithm (DES, AES, RC6 and Blowfish). This paper shows that the proposed algorithm passes the FIPS 140-2 randomness tests. After performing the encryption/ decryption steps there is no loss in the image quality. The observation is that longer keys will resist attack better than shorter keys. This scheme is secure even if everything about the system except the key is known to the users. Therefore the proposed cryptosystem is suitable for real application and transmission problems.

REFERENCES

[1] Aruljothi, S., and Venkatesulu, M., 2010, "Symmetric Key Cryptosystem based on Randomized Block Cipher", 5th International Conference on Future Information Technology, IEEE Xplore.

[2] AbdelRahman and Mohamed, H., 2003, "Crypto-Algorithms Maker kit", Proceedings of the 15th International Conference on Microelectronics, IEEE Xplore, pp. 239-242.

[3] Mihir Bellare,Shafi Goldwasser and Daniele Micciancio, 1997, "Pseudo-Random Number Generation within Cryptographic Algorithms: the DSS Case", Advances in Cryptology - Crypto 97.

[4] 2001, National Institute of Standards and Technology (NIST). FIPS-197: Advanced encryption standard.

[5] 2001, FIPS 140-2, "Security requirements for cryptographic modules", Federal Information Processing Standards publication.

[6] Nadeem, A., and Javed, M. Y., 2006, "A performance comparison of data encryption algorithm", IEEE Information and Communication Technologies, pp. 84–89,.

[7] Aruljothi, S., and Venkatesulu, M., 2013, "A New Approach To Randomized Key In The Symmetric Block Cipher Encryption Algorithm", Jokull Journal, 63(7), pp. 190-199,.

[8] 2001, FIPS 140-2, "Security requirements for cryptographic modules", Federal Information Processing Standards publication.

[9] Ji Won Yoon and Hyoungshick Kim, 2010, "An image encryption scheme with a pseudorandom permutation based on chaotic maps", Communications in Nonlinear Science and Numerical Simulations, pp.3998-4006.

[10] Elaine Barker and John M. Kelsey, 2015, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators", NIST Special Publication.

[11] Maurer, U. A., 1992, "Universal Statistical Test for Random Bit Generators", Journal of Cryptology.

[12] Andrew Rukhin et al., 2010, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", U.S. Department of Commerce / NIST. National Technical Information Service.

[13] Ozturk, I., Sogukpinar, I., 2004, "Analysis and comparison of image encryption algorithms",Transactions on engineering, Computing and Technology. pp. 1305-5313.