3D Array Block Rotation Cipher : An Improvement using shift

Sukhvinder Singh Deora[1] and Dr. Pushpa R. Suri[2]

[1] DCSA, Kurukshetra University, NC Institute of Computer Sciences

Received: 2 September 2011 Accepted: 24 September 2011 Published: 5 October 2011

Abstract

This paper on Cipher based on 3D Array Block Rotation is in continuation with our earlier paper titled A cipher based on 3D Array Block Rotation. It discusses a new rotation; lateral shift along with the earlier discussed rotation of the 3D Array block or circular shifting of plates of 3D Array in clockwise direction while enciphering and anticlockwise direction while deciphering. It also discusses the problem of relative bit positioning in the earlier specified algorithm and introduce shift rotations of the blocks as a possible solution to the problem. It uses a key of specified length which can be either transferred with the ciphertext or can be obtained by an agreed upon random bit generator. In all, it is a novel and effective cipher with good randomness property.

Index terms— Encoding, Decoding, Block cipher, randomness, Random Number Generator, 3D Array, Confusion-Diffusion, Linear Feedback Shift Rotations (LFSR), p-value

1 INTRODUCTION

ommunication involves conveying the information in one form or other to the intended receiver, using some medium. Internet, the fastest and most widely used medium of electronic information exchange, is also used for the same. However for the security of information, a technique of data encryption/ decryption is used in most of the cases.

2 II.

3 CIPHER

Cipher is a message written in a secret code. In a cryptographic system some specific units of plaintext, usually letters, are arbitrarily transposed or substituted according to a predetermined code (encoding technique) to convert it to a cipher text [1].

4 Fig.1 : Encoding

The ciphertext is then transferred over the nonsecure medium of communication and received by the receiver. The receiver then applies the decoding technique in accordance to the encoding technique to get the actual plaintext communicated to him by the sender.

5 Fig. 2 : Decoding

The basic idea behind any cryptographic algorithm is same, using confusion and diffusion to change the actual information so that it is only the intended user who can decode and understand it. Some World War II ciphers using stuttered rotors are briefly described as natural predecessors [5]. There have been algorithms like the Hill Cipher and Vernam Cipher to the DES, AES and A5 algorithms in the literature [1]. The strength of these ciphers depends upon key length, processing and the use of operations like simple negation, shift, XOR and substitution [8].

6 III. 3D ARRAY BLOCK CIPHER PROBLEM

We have developed an algorithm which encrypts/decrypts the information in the paper titled "A cipher based on 3D Array Block Rotation". We have suggested the use of Plate-wise rotation along X/Y/Z axis, at random, for the diffusion of the bits/text contained in the 3D Array.

However, we have noticed that if such a rotation is performed then there is relative bit/char positioning (red dots), equidistant from the centre (magenta dot) in case of odd sized 3D array (see Fig. 3). This relative bit/char positioning can be exploited for decoding of ciphertext produced using 3D Array Rotations as suggested in previous paper. In our current paper, we discuss introduction of circular shift operation, which will remove this positional dependency problem in the 3D Array.

7 OUR IMPROVED ALGORITHM

In this new version of our cipher based on 3D Array we are proposing the details of key length, number of rounds; which is some multiple of 8, the structure and its two kinds of rotations, the rotation policy as per the sub-keys, k-th iteration details and the overall encryption process shown through various figures and flowcharts.

a) The 3D Array Structure We are proposing that the cipher can use a key of length, 8 X Number of Rounds, minimum of 256 bits which will be sufficient to encrypt 4096 data bits. The key may be produced by using some one time pad so that there is a different ciphertext of the same plaintext each time encoding is done. The key can also be generated at the receiver end using agreed upon Random Number Generator or communicated using some highly secure algorithm before transferring the actual data. We can use a three dimensional array to store the initial plaintext. The plaintext may also be stored as row-major/column major fashion, as agreed between the sender and receiver. Considering the three axis as the axis of rotation, X, Y and Z, as shown in Fig. 4, and each layer as a rotatable plate. We can diffuse the text using clockwise rotation of 90/180/270° of particular plate at a particular axis or using linear shift rotation of the rows of the particular plate. A three dimensional matrix may be used to store the initial plaintext. There will be three possible axis of rotation as shown in Fig. 4, and axis-wise layers, as shown in Fig. 5, as a rotatable plate as seen from the three different axis of rotation X, Y and Z respectively.

8 c) Operations

Diffusion of the text can be done using clockwise rotations (see Fig. **??**) or shifting of elements of the plate (see Fig. **??**) using clockwise/circular left shift rotations of a particular plate in a particular axis of rotation as per criteria defined below. The rotations can be done using a key of 512 bits for 64 rounds cipher. We may also have variable number of rounds by taking key of appropriate number of bits. Each 8 bits from the LSB side can be used to rotate once. Consider 8 bits as shown:

9 ovember N

The Plate Number which is to be rotated is straight forward usage of the 4 bits-1, 2, 3, 4 of the subkey, whose value indicates the plate number which is to be rotated. In case of 5-6 bits to be 11, we will rotate in order of X, Y and Z axis, taking 5678 bit value number.

The Rotation policy/Number of Rotations can be decided by using 5, 6 bits of the sub-key calculated as described in Table 2: Similarly, bits 7, 8 of the sub-key can be used to decide the Axis of rotation. The two bits can be used for four possible types of selections represented by 00, 01, 10 and 11 as described in Table 3. The entire encryption process may be converted to a finite number of iterative steps. The encryption can be represented by the flowchart with niterations as shown in Fig. **??**. It uses the subkey of 8 bit length and identifies the type of rotation to be performed and then do the rotations as described in iteration detail flowchart. Next iteration is carried out on the intermediate ciphertext produced in the previous iteration. This process is repeated n number of times to complete the encryption process.

10 Fig. 8 : Encryption Process

Similarly, the decryption process is carried out exactly in the reverse manner, i.e. the n-th subkey is used first to reverse rotate the plate (in anti-clockwise direction or circular shift right rotation) and thereby obtaining the Intermediate Ciphertext (n-1). The reverse process is to be carried out for the same number of iterations with the same subkeys in reverse order as done in the encryption process. After completion of the n iterations in reverse order, we will obtain the original plaintext, refer Fig. **??**.

11 EXPERIMENTATION

In order to check the goodness of the improved cipher, we conducted lab experimentations on data in which the plaintext contained two halves, first containing all 0s and second containing all 1s. We have used Turbo C's Random Number Generator to take our keys of desired lengths. Here it is noteworthy that Turbo C's random number generator is not very good RNG. The Initial set of bits taken in the proposed ciphering technique used equal number of 0s and 1s. After enciphering using the above mentioned revised technique, we tested randomly selected 1000 bits for tests of Randomness from NIST specifications. The various tests selected for use by us vary in their importance and hardness as randomness tests [4]. Randomness in the block ciphers is considered as an important aspect of its security. One may apply various tests to ensure that cipher can work like a Random Number Generator (RNG) [7].

12 a) Monobit Test

In order to determine the number of 0s and 1s in the randomly selected bits after enciphering is approximately the same in proportion or not. If the resultant sequence becomes a random sequence, then arbitrarily selected bits must have approximately equal proportion of 0s and 1s. The focus of the test is the proportion of zeroes and ones for the entire sequence. The test assesses the closeness of the fraction of ones to ½, that is, the number of ones and zeroes in a sequence should be about the same. Result: Arbitrarily selected bits sequences from different length ciphers were selected for the Monobit tests and p-values show (Table 4) that all the tests were passed.

13 b) Frequency within a Block

Another test of randomness tests the frequency of bits within a block. The focus of this test is on proportion of 1s within M-bit blocks. The purpose of this test is to determine whether the frequency of 1s in an Mbit block is approximately M/2. Our improved algorithm gave p-values as shown (Table 5) for this test.

14 Result:

The p-values for this test also show that 100% of the random selected blocks from the encrypted sequence passed all the tests, which is a good improvement.

15 c) Run Test

The focus of this test is the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits. A run of length k consists of exactly k identical bits and is bounded before and after with a bit of the opposite value. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. The test was applied to 10 randomly selected sequences of the encrypted data for which the p-values (Table 6) below. We implemented our improved algorithm in C, on a data input of the size 16X16X16 = 163 array upto 32X32X32=323 array. Subsequences from the resultant NOTE: The code for the decrypting process will need the lastAxisOfRotationCounter value and the key. The key will be read in reverse order i.e. starting from 8 LSBs if encryption started from the 8 MSBs.

16 ovember N d) Random Excursions Test

The focus of this test is the number of cycles having exactly K visits in a cumulative sum random walk. The cumulative sum random walk is derived from partial sums after the (0,1) sequence is transferred to the appropriate (-1, +1) sequence. A cycle of a random walk random that begin at and return to the origin. The purpose of this test is to determine if the number of visits consists of a sequence of steps of unit length taken at

17 VII. ANALYSIS

The above algorithm assumes a pre-requisite of having a good unique generating function for random numbers based on a seed value, the results show that the cipher based on the 3D matrix rotation technique works good and implements confusion/diffusion technique very effectively. This 3D Block ciphering technique can be used in everyday encryption/ decryption as it is having good encrypting/decrypting efficiency too.

18 VIII. COMPLEX FORMS

The reverse computational complexity of the proposed cipher for the interceptors and intruders can be further increased by introduction of XOR round before applying rotation in case of binary input plaintext. This can be done by making use of some agreed upon random number generator which generates unique 8 bit sequences with use of a seed. The generated bits can be XORed with some selected subset of the plate under operation. This will further increase the complexity of the cipher further and will be difficult to decrypt by the interceptors.

19 IX. CONCLUSIONS

The above tests show a high rate of randomness of the bits shuffled using the improved technique. Also since the bits were initially divided in equal numbers in the two halves of the array, this shows that the cipher produces a good confusion-diffusion. It only requires an agreed upon RNG or Key for encryption-decryption. Although the new cipher can have variable number of keys used while encrypting the message, we recommend at least 2n iterations for n size array of input bits/text. [1]

[1] © 2011 Global Journals Inc. (US) Global Journal of Computer Science and Technology Volume XI Issue XIX Version I

3

Figure 1: Fig. 3 :

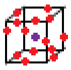
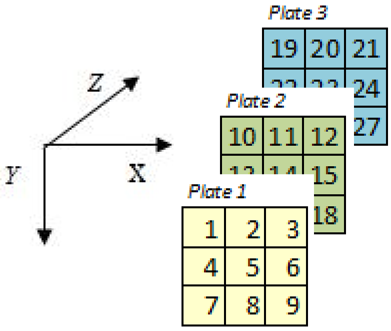
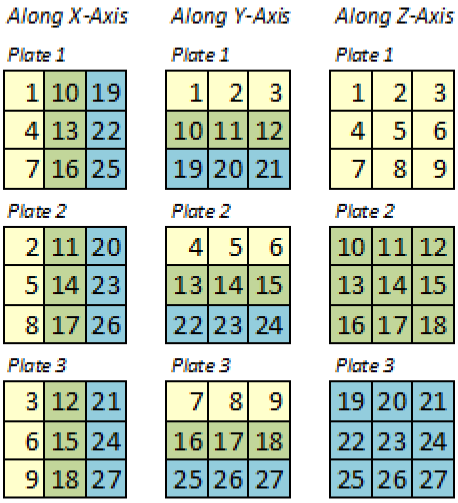


Figure 2: .



4

Figure 3: Fig. 4 :



Along X-Axis

Plate 1

1	10	19
4	13	22
7	16	25

Plate 2

2	11	20
5	14	23
8	17	26

Plate 3

3	12	21
6	15	24
9	18	27

Along Y-Axis

Plate 1

1	2	3
10	11	12
19	20	21

Plate 2

4	5	6
13	14	15
22	23	24

Plate 3

7	8	9
16	17	18
25	26	27

Along Z-Axis

Plate 1

1	2	3
4	5	6
7	8	9

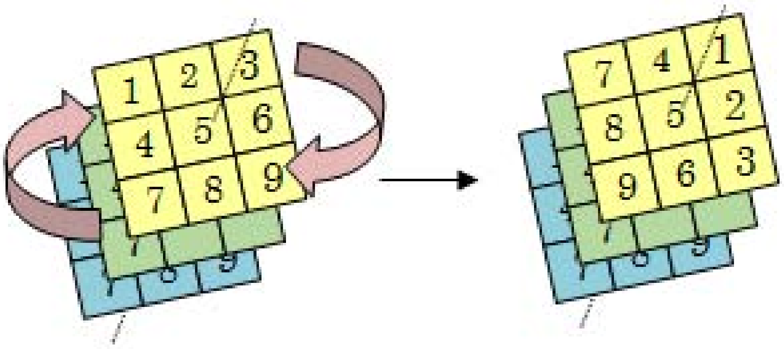
Plate 2

10	11	12
13	14	15
16	17	18

Plate 3

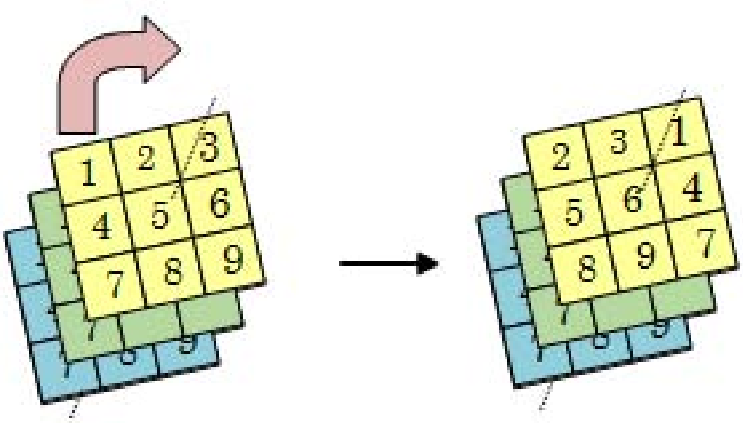
19	20	21
22	23	24
25	26	27

Figure 4:



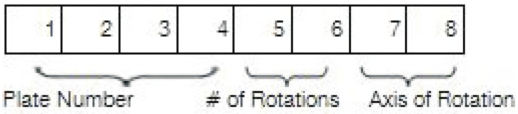
5

Figure 5: Fig. 5 :



67

Figure 6: Fig. 6 :Fig. 7 :



12

Figure 7: ? 1 2

910

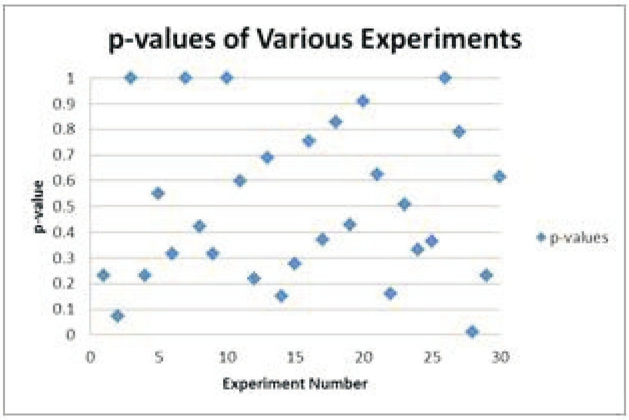


Figure 8: Fig. 9 :‘Fig. 10 :

2

Bit Value	Rotation Type	Clockwise/Shift Left Rotations
00	3D circular rotation 90°	
01	3D circular rotation 180°	
10	3D circular rotation 270°	
11	3D circular shift left rotation	The number of shifts is decided by 5678 bits combination

Figure 9: Table 2 :

3

Bit Value	Axis of Rotation
00	X
01	Y
10	Z
11	X/Y/Z in rotation starting from X axis

e) Encryption-Decryption Process

Figure 10: Table 3 :

1

Figure 11: Table 1 :

4

S No	p-value
1	0.230139
2	0.071861
3	1
4	0.230139
5	0.548506
6	0.317311
7	1
8	0.423711
9	0.317311
10	1

Figure 12: Table 4 :

5

Test Number	p-value
1	0.596677
2	0.21934
3	0.688474
4	0.14923
5	0.276677
6	0.75472
7	0.369668
8	0.829425
9	0.428474
10	0.908275

Figure 13: Table 5 :

6

Test No p-Value		Test of Randomness
1	0.622762596	PASS
2	0.161513387	PASS
3	0.505676771	PASS
4	0.333302675	PASS
5	0.363302144	PASS
6	1	PASS
7	0.790469891	PASS
8	0.011561519	PASS
9	0.230139469	PASS
10	0.613523364 PASS	

Result: All the ten tests have passed.

Figure 14: Table 6 :

Here we present the pseudo-code of the algorithm in C. The code contains the major steps to be executed in which it reads the plaintext from a file and encrypt to a ciphertext in the file named out. The algorithm was executed several times to take some arbitrary selected bit lengths for inputs to the NIST tests. //here we can put code for sliding each row of plate k of X axis for(i=0; i<size/2; i++){ for(j=i; j<size-i-1; j++){ // code for shifting the terms of each row ntimes } } } //similarly we had coded for other encryption rotation in various axis and the decryption rotation code in reverse direction to a particular state within a cycle deviates from what one would expect for a random sequence. This test is actually a series of eight tests (and conclusions), one test and conclusion for each of the states: -4, -3, -2, -1 and +1, +2, +3, +4.

.1 #include

[A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Specifications
A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Specifications, p. .

[Gollmann and Chambers (1989)] 'Clock-controlled shift registers: a review'. D Gollmann , W G Chambers . *IEEE Journal on Communications* May 1989. (7) p. 4.

[Stallings] *Cryptography and Network Security, Principles and Practices, Fourth Edition*, William Stallings . Pearson Education.

[Dhall (2010)] 'Design of a New Block Cipher Based on Conditional Encryption'. S Dhall , SK . *Seventh International Conference on Information Technology*, (Las Vegas, NV) April 2010. 714 p. .

[Lipschutz] Seymour Lipschutz . *Data Structures*, Tata McGraw Hill Publishing Company Ltd.

[Schneier] Bruce Schneier . *Applied Cryptography*, John Wiley & Sons.

[Suri (2010)] 'Sukhvinder Singh Deora: A Cipher based on 3D Array Block Rotation'. P R Suri . *IJCSNS International Journal of Computer Science and Network Security* February 2010. 10 (2) .

[Alani (2010)] 'Testing Randomness in Ciphertext of Block-Ciphers Using DieHard Tests'. Mohammed M Alani . *IJCSNS International Journal of Computer Science and Network Security* April 2010. 10 (4) .