A Randomized Encryption Scheme

Dr. Eng. / Jamal Abelfatah Morad Azzam
Dept. Head, Research Center, SCA
Ismailia, Egypt.
E-mail: jamalazzam@yahoo.com

Abstract - Cryptography is a vital part in information handling. It renders the message unintelligible to outsider by various transformations. Data cryptography is the scrambling of the content of data like text, image, audio and video to make it unreadable or unintelligible during transmission. As the data may include some sensitive information which should not be accessed by or can only be partially exposed to the general users. The principal goal guiding the design of any encryption algorithm must be security against unauthorized attacks but performance and the cost of implementation are also important concerns. In this paper, we introduce a new randomized encryption/decryption scheme. It depends on the secret key and a randomly chosen number for every encryption process. The secret key and the random number are used to generate a significant subkey for every block of data. Generation of subkeys is done by simulating a physical operation of turning a movable disk by specific values against a fixed pointer. Forwarding the random number to the receiver is not a trivial job. The algorithm shows high strength for attacks and cryptanalysis.

Keywords − encryption; decryption; ciphertext; secret key; mutation; crossover.

LIST OF SYMBOLES:

S : secret key.

SK_j : subkey blocks.

N : number of data blocks.

B_j : a data block.

MB_j : mutated data block.

CB_j : crossed over data block.

EB_j : encrypted data block

n_B, n_s : size of data block, and key block.

TA_j : turning angles of the movable disk.

R : random number.

$R^{\sim}$: compound number of R and S.

S_{mj} : modified key block.

Rm_j : modified R block.

P : pointer.

P_r : a chosen prim number.

MuCr : mutation(s) and crossover processes.

V : computed number of R and S for modulo calculations.

X_1, X_2 : fixed large numbers in encryption / decryption sides.

I: Message block number. 1,2,……., N.

$j = I$ If $I \leq 16$, and $j = I \bmod N$ if $I > 16$.

I. INTRODUCTION

Cryptography is where security engineering meets mathematics. It provides us with the tools that underlie most modern security protocols. It is probably the key enabling technology for protecting distributed systems. Cryptography refers almost exclusively to encryption, it is the process of converting ordinary information (called plaintext) into unintelligible gibberish (called ciphertext). It is also used for a variety of other information security issues including electronic signatures, which are used to prove who sent a message. Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext.

A cipher in cryptosystems is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled by the algorithm and the key. Cryptography was done to attain (CIA) i.e. confidentiality, integrity, and availability [1][2].

There are two types of cryptosystems, one-key (or symmetric key), and two-key (asymmetric key) ciphers. In symmetric key ciphers, the encryption of a plaintext and the decryption of the corresponding ciphertext are performed using the same key. Until 1976 when Diffie and Hellman introduced public-key or two-key cryptography all ciphers were one-key systems [3]. Therefore one-key ciphers are also called conventional cryptosystems. They are widely used throughout the world today, and new systems are published frequently.

There are two types of conventional cryptosystems: stream

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.171

ciphers and block ciphers. In stream ciphers, a long sequence of bits is generated from a short string of key bits, and it is then added bitwise modulo 2 to the plaintext to produce the ciphertext. In block ciphers the plaintext is divided into blocks of a fixed length, then they are encrypted into blocks of ciphertexts using the same key. The block cipher can be categorized into Feistel structure and SPN (Substitution Permutation Network) one. Feistel structure has an advantage of the same algorithm between encryption and decryption, and the feature of SPN structure is that it has a different algorithm between encryption and decryption. In particular, the SPN structure has a disadvantage that its area increases twice compared with the Feistel one when SPN structure is implemented via hardware. Except DES- 64 and TDES commonly, almost all existing algorithms require 128 bit and a variable length block cipher encryption algorithm, [4]. Block ciphers can be divided into three groups: Substitution ciphers, Transposition ciphers and, Product ciphers.

Since its introduction in 1977, the Data Encryption Standard (DES) has become the most widely applied private key block cipher [5]. Recently, a hardware design to effectively break DES using exhaustive search was outlined by Wiener [6]. AES by its turn is subjected to different cryptanalysis that presumes its ability to break AES. [7][8].

In this paper, a novel randomized scheme is proposed for block cipher. It uses the secret key and a number chosen randomly in every encryption process. Both of them are implemented to generate a different subkey for every block of message. The algorithm makes use of the concept of a physical process as a development of the scheme presented in [10]. A turning disk is divided into 2^{ns} angles. Rotating it clockwise with respect to a fixed pointer P by a specific value results a new angle value corresponding to P. Hence, this new angle is used to generate a subkey. Fig. 1a shows the original indication of P i.e., angle 0. Fig.1b shows the angle corresponding to P after turning by TA_1 = 77.

Both the secret key and the block size n_B can be of any chosen size, provided key size modulo block size is zero. In each encryption process a new random number is generated, consequently, resistance to cryptanalysis whatever it is based on differential or linear is increased [9]. Also, the large range of the random number ($R < 2^{Key\text{-}Size}$) makes brute force attacks infeasible. This scheme has significant strict avalanche effect compared to the other existing algorithms.

The rest of this paper is organized as follows: Section II addresses the proposed encryption/decryption algorithm, the generation of subkeys, the mutation and crossover processes for message blocks. Also, it explains the encryption process of the random number. Section III provides two comparative examples in comparison of strict avalanche effects with those of existing algorithms. Section IV addresses performance and analysis of the proposed algorithm in comparison with other algorithms. Section V summarizes the conclusions.

II. THE PROPOSED ALGORITH

The sizes of the secret key and data block can be chosen of any numbers. The only condition is that (key size) mod (block size) = 0. Let us consider data blocks $B_1,B_2,\ldots\ldots,B_N$, size of each is n_B = 8 bit. Key size 128 bit, 16 block × 8 bit, $S_1, S_2,\ldots..,S_{16}$, and let the prim number Pr_1 to be 131.

The turning disk is divided into 2^{128} angles. Originally, angle 0 corresponds to the fixed pointer P. Turning it by a certain number of angles e.g., TA_1 makes the value corresponding to P to be TA_1. Turning it again by TA_2, the angle corresponding to P will be TA_1 + TA_2, and so on. These angle values TA_j will be used with modified blocks of R and S i.e., Rm_j and Sm_j to generate a significant subkey for each data block B_j.

Fig. 2 shows the block diagram of the proposed algorithm, It can be summarized in the following steps :

A. Chose a Random Number $R < 2^{128}$ i.e., R lies in the range from 0 to 5.44×10^{39}. In every encryption process R is newly generated.

B. Choose a Fixed Pointer P as the value of one of the S blocks, e.g., $P = S_7$.

C. Compute the Modified Key Blocks. This operation is simply computing the modus of each separate key block S_j with a suitable number V_j. The reason of computing V_j is to strengthen the encryption process for short messages. This is done by exploiting all blocks of S and all blocks of R in V_j. For j = 1,2,……..,16.

$$S_{mj} = S_j \bmod V_j. \quad (1)$$

$$V_j = R \bmod (S' \oplus j). \quad (2)$$

$$S' = S_1 \oplus S_2 \oplus S_3 \oplus \ldots.. , S_{16}. \quad (3)$$

D. Compute the Modified R Blocks R_{mj}: Similar to S_{mj}, the modified R_j blocks of random number are computed as:

$$R_{mj} = R_j \bmod S_j. \quad (4)$$

E. Generate a Specific Values (TA_j angles) to be turned by the disk. Each block number I of the message has a specific turning angle, and consequently a specific subkey. It considers into its computation the corresponding modified random block, the modified key block, and the previous turned angle, it is computed for all message blocks (j= 1 to N) as:

$$TA_j = (TA_{j-1} + I + R_{mj})^{Smj} \bmod (P_r) \quad (5)$$

F. Compute the Subkeys: in a similar way, a subkey for each data block is computed using its turning angle get from (5), the pointer value, and the previous subkey as:

$$SK_j = (SK_{j-1} + P + TA_j)^3 \bmod (P_r) \quad (6)$$

G. Perform Mutation Process(s): at a chosen bit number in every message block, operate a self mutation process to get the mutated message blocks MB_I. The chosen bit number is the first bit in the $MB_{I,}$ then all other bits in order. In this step a multiple mutation processes can be done, each one is operated at a different bit number.

H. Cross Over Process(s): at a chosen bit number in every mutated message block, operate cross over process between each consecutive blocks of MB_I, each crossed over block CB_I is composed of two parts; first part from bit number 1 to the chosen bit number in block i, and the second part is get from the next MB_{I+1} block starting from the chosen bit number to the end.

I. Exclusive Oring: the last step in message blocks encryption is to XOR every crossed over block (CB_I) with its corresponding subkey SK_I, and with its previous encrypted one.

$$Em_I = EM_{I-1} \oplus CM_I \oplus SK_I \quad (7)$$

J. Encrypting the Random Number R: ER is impeded in the encrypted message. To be more secure to known message attacks, R is not handled itself, but a function of it e.g. $R^{\sim}$. The blocks of $R^{\sim}$ is mutated and crossed over with the blocks of a large fixed number X_1 (known to both encryption / decryption) sides. Also, another large number X_2 is used in computing $R^{\sim}$. The encrypted $R^{\sim}$ is computed as:

$$ER = R^{\sim} (MuCr)\ X_1 \quad (8)$$

Where:

MuCr: Mutation and Cross over.

$$R^{\sim} = (R + X_2) \oplus SR \quad (9)$$

$$SR_j = P^{10} \bmod (S' \oplus j) \quad (10)$$

$X_{1,}\ X_2$ are two large numbers.

K. Decryption Process: to decrypt the message, the algorithm starts by the rand number. To get $R^{\sim}$, a mutation and cross over processes to be performed first, with reference to (8) :

$$R^{\sim} = ER\ (MuCr)\ X_1 \quad (11)$$

From (10) compute SR, and then R is computed as:

$$R = (R^{\sim} \oplus SR) + X2 \quad (12)$$

After getting the random number the decryption process goes on as encryption steps.

III. COMPARATIVE EXAMPLES

Two examples are explained here, each example encrypts two messages. The difference between the two messages is 1 bit flipped, the results show the avalanche effect of the algorithm.

A. Example 1:

The secrete key is: EXAMPLES

In decimal: S1= 69, S2 = 88, S3 = 65, S4 = 80, S5 = 76, S6 = 69, S7 = 83, S8 =13, S9 = 10.

In binary: 0100,0101 0101,1000 0100,0001 0101,0000 0100,1100 0100,0101 0101,0011 0000,1101 0000,1010.

Message 1: NET<u>W</u>ORK ≡ 0100,1110 0100,0101 0101,0100 0101,011<u>1</u> 0100,1111 0101,0010 0100,1011 0101,0011.

Message 2: NET<u>V</u>ORK ≡ 0100,1110 0100,0101 0101,0100 0101,011<u>0</u> 0100,1111 0101,0010 0100,1011 0101,0011.

Message block size is 8 bit. Number of message blocks N = 8, Secret key size 72 bit, 8 bit × 9 block.

Let the random number R = 1000 ≡ 0000,0011 1110,0011, and the prime number Pr = 131.

Choose the pointer P to be S7, so P = 83 ≡ 0101,0011.

-The modified subkeys are $Sm_J = S_J \bmod 32$.

$Sm_1 = 5$, $Sm_2 = 24$, $Sm_3 = 1$, $Sm_4 = 16$, $Sm_5 = 12$, $Sm_6 = 5$, $Sm_7 = 19$, $Sm_8 = 13$, $Sm_9 = 10$.

-The modified random blocks: $Rm_I = R \bmod S_J$.

$Rm_1 = 34$, $Rm_2 = 32$, $Rm_3 = 25$, $Rm_4 = 40$, $Rm_5 = 12$, $Rm_6 = 34$, $Rm_7 = 4$, $Rm_8 = 12$, $Rm_9 = 0$.

-The turning angles for each block: $TA_J = (TV_{J-1} + I + Rm_J)^{SmJ} \bmod (P_r)$.

$TA_1 = 45$, $TA_2 = 39$, $TA_3 = 67$, $TA_4 = 15$, $TA_5 = 45$, $TA_6 = 71$, $TA_7 = 108$, $TA_8 = 128$, $TA_9 = 113$.

-The subkeys are : $SK_j = (SK_{j-1} + P + TV_J)^3 \bmod (P_r)$:

$SK_1 = 104 \equiv 0110,1000$, $SK_2 = 111 \equiv 011,1111$,
$SK_3 = 130 \equiv 1000,0010$, $SK_4 = 11 \equiv 0000,1011$,
$SK_5 = 119 \equiv 0111,0111$, $SK_6 = 21 \equiv 0001,0101$,
$SK_7 = 105 \equiv 0110,1001$, $SK_8 = 2 \equiv 0000,0010$,
$SK_9 = 118 \equiv 0111,0110$, $SK_{10} = 96 \equiv 0110,0000$,
$SK_{11} = 97 \equiv 0110,0001$, $SK_{12} = 31 \equiv 0001,1111$.

-Mutation Process of Message Blocks: At arbitrary bit number perform self mutation (bit number 4 is chosen here) as follows: First block of message is M_I = ***0100***,1110 m will be MM_1= 1110,***0100***. With the same procedure: MM_2 = 0101,0100, MM_3 = 0100,0101, MM_4 = 0111,0101,MM_5 = 111,0100, MM_6 = 0010,0101, MM_7 = 1011,0100, MM_8 = 0011,0101.

-Crossover Process:

At another arbitrary bit number perform cross over processes between adjacent mutated blocks (bit number 4 is chosen here for simplicity) as:

MM_1 (1110, 0100) to be crossed over with MM_2 = 0101,0100, to get CM_1 = 1110,0100, and

CM_2 = 0101,0101, CM_3 = 0100,0101, CM_4 = 0111,0100, CM_5 = 1111,0101, CM_6= 0010,0100, CM_7 = 1011,0101, CM_8 = 0011,0100.

-Exclusive Oring:

To encrypt the crossed over blocks, XOR it with the previous one and its corresponding subkey,

$$EM_I = EM_{I-1} \oplus CM_I \oplus SK_I.$$

The encrypted message (NETWORK) is:

1000,1100, 1011,0110, 0111,0001, 0000,1110,
1000,1100, 1011,1101, 0110,0001, 0101,0111.

Applying the same procedure for the second message (NETVORK) we get:

1101,0001, 1101,1111, 0011,0110, 1011,1100,
0110,0011, 1110,0111, 0101,0111, 1101,0011.

The total message length is 64 bits. By flipping one bit in the message, number of bits changed in the encrypted message is 34 bit. This means flipping one bit in the message causes 53.1 %. of encrypted message bits to be flipped.

The same example was carried out by other algorithms [11], and the results are shown in Table I. It shows number of flipped bits FB and its percentage.

B. Example 2:

The input plaintext is " DISASTER".

Flipping one bit from the plaintext, we get "DISCSTER", (one flipping A (01000001) to C (01000011)).

The Key used is "SRIRAMSR".

DISASTER encrypted message is:

00111,11011 10001,10100 10101,01000 10100,10011
11011,01001 01001,11011 11011,00111

DISCSTER encrypted message is:

01010,00110 00100,01011 01100,11100 11110,10000
00100,10111 01101,01110 11111,10010

Number of flipped bits in the encrypted message is 42 bit (out of 65 bits of the original message).

Avalanche effect = 42 * 100 / 65 = 64.6%.

The same example was carried out by other algorithms [12], and the results are shown in Table II.

TABLE I COMPARISON OF AVALANCH EFFECT

OF EXAMPLE 1

SN	Encryption Techniques	ECB		CBC		CFB		OFB	
		FB	%	FB	%	FB	%	FB	%
1	DES	33	51.5	34	53.1	20	31.2	1	1.5
2	AES	69	53.9	66	51.5	16	25	1	4.5
3	BLOFISH	34	53.1	31	48.4	20	31.2	1	1.5
						FB		%	
4	**The Proposed Algorithm**					34		53.1	

IV ANALYSIS AND PERFORMANCE

The algorithm is tested, results are compared with known existing algorithms, the summery of its performance and analysis is explained hereinafter.

A. Performance

The algorithm runs on a 3.2 GH PC with different lengths messages, the concluded speed is 5.19 cycle per byte (587 MiB/s) for the encryption process, which is very fast compared to different algorithms shown in the Table III benchmark, [13].

Number of overhead bits is fixed and irrelevant to the message length (double the key size) i.e., 128*2 bits only for random number mutated and crossed over with X_1).

B. Analysis

Although the algorithm does not depend on substitution permutation networks (SPNs), it keeps its cryptographic static and dynamic prosperities. Its strict avalanche effect causes 64.6 % of bits in average to be flipped in the enciphered text for one bit flipped in plaintext as shown in the Table 1.

TABLE II COMPARISON OF AVALANCH EFFECT

For EXAMPLE 2

Encryption Technique	**No. of flipped bits**	**%**
Playfair Cipher	4	6.25
Vigenere Cipher	2	3.13
Caesar Cipher	1	1.56
DES	35	54.68
Blowfish	19	28.71
The Proposed technique	42	64.6

Strict avalanche criterion is a measure of a cipher's randomness. High randomness ensures the algorithm resistance to statistical, clustering, linear, and differential cryptanalysis. So that, when encrypting the same message several times, it will produce different cipher texts each time.

Key size can be of any chosen number. Consequently, block size can be larger, e.g., key size of 1024 bit with block 128 or 256 or 512, also, the larger the key size the larger range for the random number. $0 < R < 2^{\text{key-size.}}$. Consequently, attacking the algorithm become harder.

The algorithm has two different arbitrary bit numbers (from 1 to block_size -1), one for mutation and the other for crossover process. Also, number of (mutation then crossover) rounds can be increased to any chosen number. In this case each mutation process can be done at a different bit number. Another list of bit numbers can be used for crossover processes.

The number of brute force trials in the worst case is 9.7×10^{89} or 1.02×10^{71} years (assuming 10^{10} decryption process per second). In a known message attack, if the attacker knows both plain and ciphered messages completely, he cannot get R because it is not send, but a functions of it is used. To get the function of R, i.e., $R^{\sim}$, the attacker has to make reverse mutation and crossover in 8^8 operations, (nine possible mutation bit numbers, and for each one nine possible crossover bit number). While guessing R from $R^{\sim}$ and X_2, there are 2^{128} possible changes in $R^{\sim}$. So, the attacker needs 4.6×10^{911} or 1.45×10^{849} years, (assuming 10^{10} decryption process per second).

The algorithm time and space complexity is O (n), i.e., the required time and space for encryption/decryption increases linearly with message length. However attacker algorithm is NP complete

TABLE III ALGORITHM SPEED COMPARISON

Algorithm	**MiB/Second**	**Cycles Per Byte**
AES/GCM (2K tables)	102	17.2
AES/GCM (64K tables)	108	16.1
AES/CCM	61	28.6
AES/EAX	61	28.8
CRC32	253	6.9
Adler32	920	1.9
MD5	255	6.8
DES/CTR	32	54.7
DES-XEX3/CTR	29	60.6
DES-EDE3/CTR	13	134.5
PROPOSED ALGORITHM	590	5.17

V. CONCLUSIONS

In this paper, a randomized, novel, and immune scheme for block cipher encryption is introduced. The scheme shows high strength of confidentiality even for known message attack. In the proposed scheme, both the key length and the block size can be of any chosen size, provided key length modulo block size is zero.

The algorithm has a high degree of randomness; its strict avalanche effect is very significant and surpasses a lot of the famous algorithms. This proves its immunity to cryptanalysis. Knowing the algorithm, the plaintext and the ciphered text does not reveal useful information for the attacker to crack the key or the random number, because in every run of the algorithm a new random number and new subkeys are generated, consequently different ciphered texts for the same plaintext.

The algorithm has a strong strict avalanche criterion (SAC) (65% in average). Also, it keeps the cryptographic static properties of substitution permutation networks (SPNs) of completeness, nonlinearity. In the same time the algorithm provides perfect security, and every bit in the information message is encrypted using a different subkey.

Implementing random numbers of a large range ($2^{\text{key\_size}}$) in encryption, makes cracking it is infeasible. For the chosen length (128 bit), brute force attack needs 1.02×10^{71} Years to crack that

key, (assuming 10^{10} check per second). Also, it is safely distributed between sender and receiver in a new procedure.

As an additional feature of the scheme some of its parameters are selective and not fixed e.g., Key size – Block size – Bit number to perform mutation – Bit number to perform crossover – number of mutation then crossover rounds.

An attractive feature of the algorithm is that, its time and space complexity is O (n). So, the required memory resources and computation time are not increased in a large scale with the increase of message length. In the same time the attacking algorithm is NP complete.

REFRENCES

[1] W. Stallings, *cryptography and network security: principles and practices*, 5th ed., Prentice Hall.

[2] "Cryptology (definition)", http:// www.marriam-webster.com/dictionary/cryptology, Merriam- Webster's Collegiate Dictionary (11th edition Ed.).Merriam- dictionary / cryptology, retrieved, 03-25-2015.

[3] H. Beker and F. Piper, "Cipher Systems: The Protection of Communications", John Wiley & Sons, New York, 1982.

[5] National_ Bureau_ of_ Standards,“ Data Encryption Standard (DES)”, federal Information Processing Standard Publication FIPS PUB 46-3,U.S. dept. of commerce/national institute of standards and technology, 1977.

[6] I. Ben Aroya and E. Biham, " Differential cryptanalysis of Lucifer". In D.R. Stinson, editor, Advances in Cryptology: CRYPTO'93, LNCS 773, 1993.

[7] D. Warren,"1. AES seems weak. 2. Linear time securecryptography",http://www.researchgate.net/publication/220335792_1._AES_seems_weak._2._Linear_time_secure_cryptography, IACR Cryptology ePrint Archive 01/ 2007, retrieved 2015-03-01.

[8] E. Biham and A. Shamir, " Differential Cryptanalysis of DES-like Cryptosystems", Journal of Cryptology, vol. 4, no. 1, pp 3-72,1991.

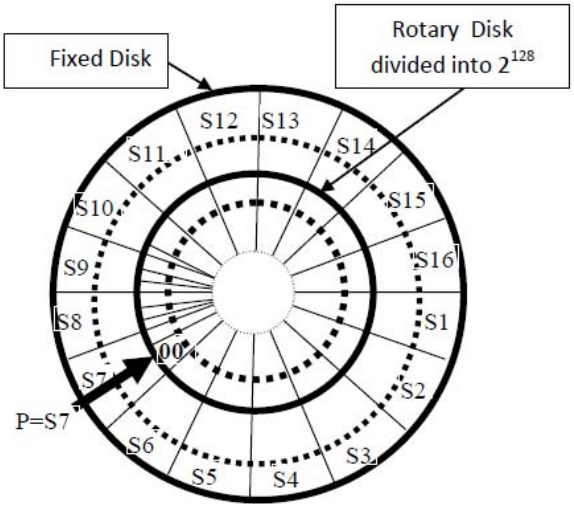
[9] L. Brown, J. Pieprzyk, and J. Seberry, "LOKI - a cryptographic primitive for authentication and secrecy applications". In J. Seberry and J. Pieprzyk, editors, Advances in Cryptology: AusCrypt'90, LNCS 453,. Springer Verlag, 1990.

[10] Jamal Azzam, "Enhanced Authenticated Encryption Scheme", *The Ninth International Conference on Emerging Security Information, Systems and Technologies*, Venice, Italy, August 23 - 28, 2015.

[11] Nikata and Ranjeet Kaur, "A survey ON Secret Key Encryption Techniques",International Jornal of Research in Engineering & Te chnology, Vol. 2, May 2014.

[12] Sriram Ramanujam and Marimuthu Karuppiah, "Designing an algorithm with high avalanche effect", IJCSNS International Journal of Computer Science and Network Security, VOL. 11 NO.1, Jan, 2011.

[13]http://www.cryptopp.com/benchmarks-p4.html, accessed at sept. ,16, 2015.



. Figure 1-A. Initial Position of a Moving Disk.

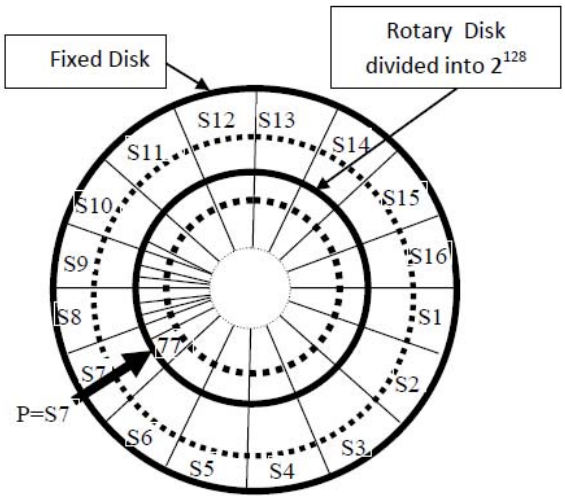


Figure 1-B. Same Disk After Rotating by Angle =77.

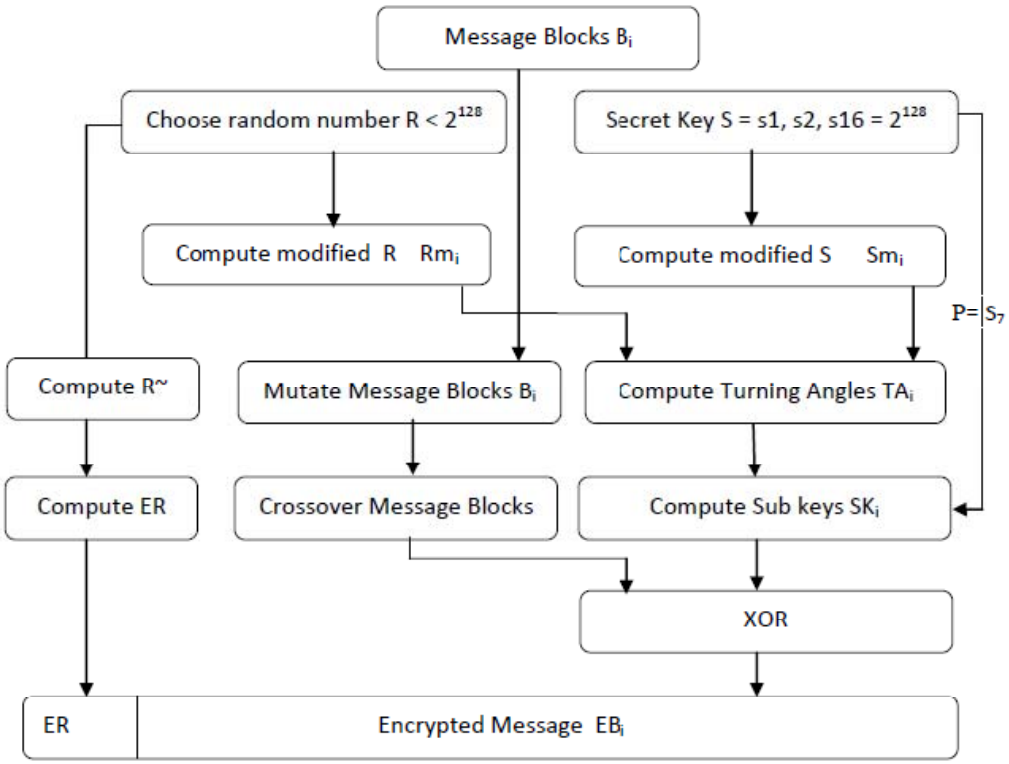


Figure 2 Block Diagram of the Proposed Algorithm.