D. Sravan Kumar, CH. Suneetha, A.Chandrasekhar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 1, Jan-Feb 2012, pp. 417-425

A Key-Scheduled Block Cipher Using element-wise Linear Transformation and Logical XOR Operation

[1]D. Sravan Kumar [2]CH. Suneetha [3]A.Chandrasekhar

[1]Reader in Physics, SVLNS Government College, Bheemunipatnam, Visakhapatnam Dt., India
[2]Assistant Professor in Engineering Mathematics, GITAM University, Visakhapatnam, India
[3]Professor in Engineering Mathematics, GITAM University, Visakhapatnam, India

Abstract:-
Cryptography is a key technology in electronic security systems. Modern cryptographic techniques have many uses, such as to digitally sign documents, for access control, to implement electronic money, and for copyright protection. The increased use of computer and communication systems by industry has increased the risk of theft of proprietary information. In general, cryptographic primitives are designed to satisfy particular security objectives which may be built from the four basic objectives – confidentiality, data integrity, authentication and non-repudiation. In the present paper a new key-scheduled block cipher is proposed using element-wise Linear Transform and logical XOR operation.

Key Words: - Linear Transformation; Logical XOR operation; Encryption; Decryption

Introduction:-

Encryption is the process of converting a plaintext message into cipher text which can be decoded back into the original message using the secret key. An encryption/ decryption algorithm, along with a key is used in the encryption/decryption of data. There are several types of data encryption schemes which form the basis of network security. Encryption schemes are generally based on either block or stream ciphers. Historically the focus of encryption has been on the use of symmetric encryption to provide confidentiality. It is only in the last several decades that other considerations, such as authentication, integrity, digital signature have been included in the theory and practice of cryptology. The security of the message basically depends on two factors 1) confidentiality and 2) authentication. One of the means of achieving confidentiality of the message is encrypting bulk digital data using block ciphers. Single round of encryption offers inadequate security but multiple rounds of encryption offer increasing security. In the present paper we propose a new method of encryption of data block in 8 rounds using Linear Transformation and Logical XOR operations with a one-time sub key derived for each round of encryption from the session key of that particular data block which will be generated from the master key (secret key). Several researchers [10, 15] in cryptography used linear transformation in their encryption algorithms. In the key scheduled algorithm [1, 2] for encryption/decryption of the data proposed in this paper, the size of the data block is selected to be of 64 characters. The characters of each data block are coded to Hexadecimal numbers using ASCII code table and are written as an 8x8 matrix row-wise. Each Hexadecimal element is encrypted in 8 rounds using Linear Transformation operation so that the outcome is a new element. The key used in the Linear Transformation operation is different for different elements in each round of encryption i.e. the Linear Transformation is not fixed, but depends on the sub key derived for each round of encryption from the session key of that particular data block. The session key of each data block is generated [12, 13] from the master key (secret key) agreed upon by the communicating parties. Between two successive Linear Transformation operations the logical XOR operation is performed on each element of the matrix with its nearest four neighboring elements so that on completion of eight rounds of Linear Transformation operations and XOR operations good avalanche effect is achieved which is one of the desired properties of encryption algorithm. The procedure designed in this method ensures that the message is highly secure as long as the key selected by the communicating parties is secure. The encryption/decryption procedure further assures relatively low computation overhead. A Logical XOR gate is digital logical gate which performs a logical operation on one or more logic inputs and produces a single logic output. Several researchers of cryptography used the logical operation XOR [7,8] in their encryption protocols.

For describing the algorithm the following notation and definitions are adopted:-

Symbols and Notation:-

Symbol	Expression	Meaning
M,m,n,l	${}^{l}M_{n}^{m}$	The 8x8 matrix whose elements are Hexadecimal Numbers, where l,m,n take integer values, n∈N l, m ∈ {0,1,2……8]
K,m,n	K_{n}^{m}	The 8x8 matrix whose elements are the Hexadecimal codes of ASCII characters **excluding the null character**
R,E, ∧	$\hat{L}_{E}$	Encryption Operator using Linear Transformation
R,D, ∧	$\hat{L}_{D}$	Decryption Operator using Linear Transformation
X,E, ∧	$\hat{X}_{E}$	The logical XOR operator used in the encryption process
X,D, ∧	$\hat{X}_{D}$	The logical XOR operator used in the decryption process
A	A = {1,3,5,7,9,B,D,F}	A is the set of all numbers which are relative primes to 16, used for obtaining the principal key matrix for the encryption of the matrix ${}^{l}M_{n}^{m}$ using Linear Transform
P, ∧	$\hat{P}$	The operator used to obtain the principal key matrix from the sub-key used in that particular round
P,n,m	P_{n}^{m}	The Principal Key matrix obtained from K_{n}^{m}
S, ∧	$\hat{S}_{n}^{m}$	The operator used for deriving the sub key for the m^{th} round of encryption from the key used for the first round encryption of n^{th} data block
G, ∧	$\hat{G}_{n}$	The operator used for generating the session key for the encryption of the n^{th}data block from the session key used in the encryption of the $(n-1)^{th}$data block.
M,l,m,n,i,j	$\left[{}^{l}M_{n}^{m}\right]_{ij}$	Represents the element in the i^{th} row and j^{th} column of the matrix $\left[{}^{l}M_{n}^{m}\right]$

Definitions:

1) ${}^{l}M_{n}^{m}$ denotes an 8x8 matrix whose elements are Hexadecimal numbers. The right superscript m denotes the number of times the linear transformation operator $\overset{\Lambda}{L}_E$ is performed on the elements of the matrix M. The left superscript l represents the number of times the logical XOR operator $\overset{\Lambda}{X}_E$ is applied on the matrix M. The right subscript n indicates the number of data block that is being encrypted.

2) K_n^m represents an 8x8 matrix whose elements are the Hexadecimal numbers **excluding the null character** which is called the key matrix used in the encryption operation of the $(m+1)^{th}$ round of the nth data block matrix ${}^{m}M_{n}^{m}$ using Linear Transformation. The right superscript m of K represents the number of round of encryption using the linear transformation on the matrix ${}^{m}M_{n}^{m}$, where m ranges from 0 to 7. The right subscript indicates the data block which is being encrypted.

3). P_n^m is the Principal Key matrix obtained from the main key matrix K_n^m for $(m+1)^{th}$ round of encryption of nth data block using Linear Transform.

Let $(K_n^m)_{ij} = K_1K_0$, where K_1, K_0 are the Hexadecimal digits in 16' place and 1's place.

$(P_n^m)_{ij} = P_1P_0$ Where

$P_0 = K_0, P_1 = X$ if $X \in A$, else $X+1$

where $X = (K_1K_0 + K_1 + K_0)_{\text{mod16}}$

$\overset{\Lambda}{P}(K_n^m) = P_n^m$

4). The Linear Transformation used in the algorithm is of the form

$X = (ax+b)_{\text{mod16}}$ where $a \in A$ and b may be any hexadecimal from 0 to F.

Then x can be obtained by the inverse linear transform

$x = [a^{-1}(X + b^{-1})]_{\text{mod16}}$

For all $a \in A$, a^{-1} is defined in the following table 1 and b^{-1} is defined in the table 2

TABLE 1

a	1	3	5	7	9	B	D	F
a^{-1}	1	B	D	7	9	3	5	F

TABLE 2

b	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
b^{-1}	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1

5). The Operator $\overset{\Lambda}{L}_E\left[{}^mM_n^m, P_n^m\right]$ is the operator that affects Linear Transformation. It has two arguments ${}^mM_n^m$ and P_n^m. The first argument is the matrix on which the operation $\overset{\Lambda}{L}_E$ is applied. The second argument is the key matrix used for performing the operation $\overset{\Lambda}{L}_E$ in the $(m+1)^{th}$ round encryption of n^{th} data block matrix. With this operation the right superscript m of the matrix ${}^mM_n^m$ increases by one unit. With this operation each hexadecimal element $\left[{}^mM_n^m\right]_{ij}$ of the matrix ${}^mM_n^m$ is encrypted using the Linear Transformation

Let = $\left[{}^mM_n^m\right]_{ij} = M_1M_0$ which is a Hexadecimal number. M_1M_0 is encrypted as

$$M_1^E = (M_1P_1 + P_0)_{\text{mod}16}$$
$$M_0^E = (M_0P_1 + P_0)_{\text{mod}16}$$

Where P_0 and P_1 can be obtained as in Definition (3).

For example let $\left[{}^mM_n^m\right]_{11} = 7F$ and let $P_1 = 5$ and $P_0 = 3$ then 7F is encrypted as

$$(7.5+3)_{\text{mod}16} = 6 \text{ and } (F.5+3)_{\text{mod}16} = E$$

So, 7F is encrypted as 6E

$$\overset{\Lambda}{L}_E\left[{}^mM_n^m, P_n^m\right] = {}^mM_n^{m+1}$$

All the elements of the matrix ${}^mM_n^m$ are encrypted using the Linear Transform and are written in the form of 8 bit binary numbers.

6). $\overset{\Lambda}{X}_E[{}^mM_n^{m+1}]$ represents the XORing of each element of the matrix ${}^mM_n^{m+1}$ with its nearest four neighboring elements in the encryption process. With this operation the left superscript m increases by one unit. With this operation each element $\left[{}^mM_n^{m+1}\right]_{ij}$ of the matrix ${}^mM_n^{m+1}$ which is in the 8 bit binary format is XORed with the nearest four neighboring elements which results in the matrix ${}^{m+1}M_n^{m+1}$

$$\overset{\Lambda}{X}_E[{}^mM_n^{m+1}] \rightarrow ((((({}^mM_n^{m+1}{}_{ij} XOR {}^mM_n^{m+1}{}_{ij-1}) XOR {}^mM_n^{m+!}{}_{i+1,j}) XOR {}^mM_n^{m+1}{}_{ij+1}) XOR {}^mM_n^{m+!}{}_{i-1j})$$

$$\overset{\Lambda}{X}_E[{}^{m}M_n^{m+1}] = {}^{m+1}M_n^{m+1}$$

Example $\left[{}^{m}M_n^{m+1}\right]_{24} = 10010100, \left[{}^{m}M_n^{m+1}\right]_{23} = 10100010$ $, \left[{}^{m}M_n^{m+1}\right]_{34} = 00111000$ $\left[{}^{m}M_n^{m+1}\right]_{25} = 10010010, \left[{}^{m}M_n^{m+1}\right]_{14} = 01100101$ then

$$\overset{\Lambda}{X}_E\left[{}^{m}M_n^{m+1}\right]_{24} = ((((10010100 XOR 10100010) XOR 00111000) XOR 10010010) XOR 01100101)$$

$$\left[{}^{m+1}M_n^{m+1}\right]_{24} = 11111001$$

All the elements of the matrix ${}^{m+1}M_n^{m+1}$ which are the 8 bit binary numbers are then converted into Hexadecimal numbers

7). $\overset{\Lambda}{X}_D[{}^{m}M_n^{m}]$ represents the XORing of each element of the matrix ${}^{m}M_n^{m}$ with the surrounding elements in the decryption process. With this operation the left superscript m decreases by one unit. All the elements of the matrix ${}^{m}M_n^{m}$ are XORed with the nearest neighboring elements as

$$\overset{\Lambda}{X}_D[{}^{m}M_n^{m}] \rightarrow ((((({}^{m}M_{n\,ij}^{m} XOR {}^{m}M_{n\,i-1j}^{m}) XOR {}^{m}M_{n\,ij+1}^{m}) XOR {}^{m}M_{n\,i+1j}^{m}) XOR {}^{m}M_{n\,ij-1}^{m}).$$

$\overset{\Lambda}{X}_D[{}^{m}M_n^{m}] = {}^{m-1}M_n^{m}$ where m ranges from 0 to 7.

$\left[{}^{m}M_n^{m}\right]_{45} = 11111001, \left[{}^{m}M_n^{m}\right]_{35} = 10100010$ $\left[{}^{m}M_n^{m}\right]_{46} = 00111000$ $\left[{}^{m}M_n^{m}\right]_{55} = 10010010$ $\left[{}^{m}M_n^{m}\right]_{44} = 01100101$ then

$$\overset{\Lambda}{X}_D\left[{}^{m}M_n^{m}\right]_{45} = ((((11111001 XOR 01100101) XOR 10010010) XOR 00111000) XOR 10100010)$$

$$\left[{}^{m-1}M_n^{m}\right]_{45} = 10010100$$

All the elements of the matrix ${}^{m-1}M_n^{m}$ which are in the 8 bit binary format are written as Hexadecimal numbers.

8). The Operator $\overset{\Lambda}{L}_D\left[{}^{m-1}M_n^{m}, P_n^{m}\right]$ is the decryption operator using Linear Transformation. It has two arguments ${}^{m-1}M_n^{m}$ and P_n^{m}. The first argument is the matrix on which the operation $\overset{\Lambda}{L}_D$ is performed in the (9-m)th round of decryption of the nth data block matrix. The second argument P_n^{m} is the key matrix used for performing the Linear Transformation operation $\overset{\Lambda}{L}_D$ in the mth round decryption of the nt^{h} data block matrix. With

this operation the right superscript m of the matrix ${}^{m-1}M_n^m$ decreases by one unit. With this operation each Hexadecimal element $[{}^{m-1}M_n^m]_{ij}$ of the matrix ${}^{m-1}M_n^m$ is decrypted using the Linear Transformation

Let $[{}^{m-1}M_n^m]_{ij} = M_1^E M_0^E$ then M_1 and M_0 can be obtained by taking the inverse linear transformation as

$$M_1 = [P_1^{-1}\{M_1^E + P_0^{-1}\}]_{\text{mod16}}$$

$$M_0 = [P_1^{-1}\{M_0^E + P_0^{-1}\}]_{\text{mod16}}$$

where P_1^{-1}, P_0^{-1} can be obtained from table1 and table 2

For example if $[{}^{m-1}M_n^m]_{11}$ = 6E then 6E is decrypted as

$$[D\{6+13\}]_{\text{mod16}} = 7$$
$$[D\{E+13\}]_{\text{mod16}} = F$$

Where $P_1^{-1} = D$ and $P_0^{-1} = D$

$$\overset{\Lambda}{L}_D\left[{}^{m-1}M_n^m, K_n^m\right] = {}^{m-1}M_n^{m-1}$$

All the elements of the matrix ${}^{m-1}M_n^{m-1}$ which are Hexadecimal numbers are converted into Binary numbers.

9). $\overset{\Lambda}{S}_n^m$ is the operator used for deriving the sub key for the (m+1)th round encryption of nth data block from the session key used for the first round operation i.e.

$$\overset{\Lambda}{S}_n^m[K_n^0] = K_n^{m-1}$$

The key matrix K_n^{m-1} for the mth round encryption of the nth data block is obtained from K_n^0 by shifting the columns of the matrix K_n^0 to the right by (m-1) places.

i.e. $\left[K_n^{m-1}\right]_{ij} = \left[K_n^0\right]_{ij-m+1}$

10). $\overset{\Lambda}{G}_n$ is the operator which defines the session key generation for the first round encryption of the nth data block from the session key used for the first round encryption of the (n-1)th data block.

$$\overset{\Lambda}{G}_n\left[K_{n-1}^0\right] = K_n^0, where\left[K_n^0\right]_{ij} = \left[(K_{n-1}^0)_{ij} + (K_{n-1}^0)_{ij+1}\right]_{\text{mod8}}$$

i and j take values in all the definitions made above from 0 to 7. If i+1 or i-1 or j+1 or j-1or any subscript fall out of the range {0,1,2,….7} then modulo 8 of that number be considered. The session key of each data block is itself sub key for the first round of encryption of the data block.

Before communicating the messages both the sender and the receiver agree upon to use the secret key which is in the form of an 8x8 matrix K (master key) whose elements are the Hexadecimal numbers **excluding null character**. This matrix K (master key) is denoted by K_1^0 in the encryption/ decryption process i.e. the master key itself is the session key for the encryption/decryption of first data block. For implementing the algorithm the entire message is divided into data blocks $D_1,D_2,D_3\ldots\ldots D_n$ of 64 characters each where n is a natural number. The characters in each message block are Hex coded using ASCII code table and are arranged in the form of 8x8 matrices ${}^0M_1^0, {}^0M_2^0, {}^0M_3^0, \ldots\ldots {}^0M_n^0$ row wise. The number of characters in the message always may not be the integral multiple of 64. Hence, at the end of the message the sender adds three # characters (###) and ensures that the message fills integer number of text blocks by adding random different characters after the three # characters.

Algorithm :-

Encryption-

Set n = 1

m = 1

$$K_1^0 = K$$

Step 1:- PRINT DATA BLOCK = n

Step2:- PRINT ENCRYPTION ROUND = m

Step3:- $S_n^m[K_n^0] = K_n^{m-1}$

Step4:- $\overset{\Lambda}{P}[K_n^0] = P_n^{m-1}$

Step5:- $\overset{\Lambda}{L}_E\left[{}^{m-1}M_n^{m-1}, K_n^{m-1}\right] = {}^{m-1}M_n^m$

Step6:- Convert all hexadecimal elements of ${}^{m-1}M_n^m$ to 8 bit binary numbers

Sep7:- $\overset{\Lambda}{X}_E[{}^{m-1}M_n^m] = {}^mM_n^m$

Step8:- Convert all the elements of ${}^mM_n^m$ which are 8 bit binary numbers to hexadecimal numbers

Step9:- If m < 8, increment m by one unit and go to Step 2

Else set m = 1

If n < N increment n by one unit and go to Step 1.

Else Stop

After eight rounds of encryption using Linear Transformation and Logical XOR operations all the hexadecimal entries of each data block are converted to text characters using ASCII code table and is communicated to the receiver as the cipher text.

The receiver after receiving the cipher text divides into data blocks $D_1^E, D_2^E, D_3^E .....D_n^E$ of 64 characters each. All the 64 characters of each data block are converted to 8 bit binary numbers using ASCII code table and are arranged in the form of 8x8 matrices ${}^8M_1^8, {}^8M_2^8, {}^8M_3^8, ... {}^8M_n^8$

Decryption:-

Set n = 1

m = 8

$K_1^0 = K$

Step 1:- PRINT DATA BLOCK = n

Step2:- PRINT DECRYPTION ROUND = 9-m

Step3:- $S_n^m[K_n^0] = K_n^{m-1}$

Step4:- $\overset{\Lambda}{P}[K_n^0] = P_n^{m-1}$

Step5:- $\overset{\Lambda}{X}_D[{}^mM_n^m] = {}^{m-1}M_n^m$

Step6:- convert all the elements which are in 8 bit binary format to hexadecimal numbers

Step7:- $\overset{\Lambda}{L}_D\left[{}^{m-1}M_n^m, K_n^{m-1}\right] = {}^{m-1}M_n^{m-1}$

Step8:- convert all the elements which are hexadecimal number to 8 bit binary numbers

If m > 0, decrement m by one unit and go to Step 2

Else set m = 8

If n < N increment n by one unit and go to Step 1.

Else Stop

After 8 rounds of decryption of all the cipher text blocks using Logical XOR and Linear Transformation the original message can be obtained.

Security Analysis:- In the key scheduled algorithm proposed here different keys are used for encrypting different data blocks which are called session keys generated from the master key (secret key) between the sender and the receiver) and the key used for the encryption of each round is different and is derived from the session key of the corresponding round which is called the sub key. As different keys are used for different data blocks cipher is less vulnerable to passive attacks. As each element of the message matrix M is encrypted using Linear Transformation with different keys (not fixed key) and logical XOR operation is performed with its all nearest neighboring elements the same characters in the plain text space are mapped to different characters of the cipher text space even though they are in the same text block or different text blocks. So, cipher text is not easily amenable to cryptanalysis [5,7]. Even the change of a single element of the message matrix changes almost the entire cipher block matrix, i.e., to say that the proposed algorithm has achieved a good avalanche effect [4,6] which is one of the desired qualities of a good encryption algorithm.

If the same message is sent in I and II (or any subsequent) data block, they are mapped to different cipher texts, i.e., even if the same message is sent repeatedly in the same message block, the messages are enciphered to different cipher texts. Hence, active attacks such as chosen plain text attacks [9,11], chosen cipher text attacks [14] are quite difficult to execute. Hence, the proposed algorithm is less vulnerable to active attacks. The present encryption algorithm is at most secure against man-in-middle attack [3] because the entire master key is agreed upon by the sender and the receiver rather than the electronic exchange of the parts of the key.

The proposed key scheduled algorithm in this paper is less prone to timing attacks because the time required to encipher or decipher a data block is same for all data blocks since time for enciphering or deciphering is independent of characters in the data block. Even though the original message contains less that 64 characters the remaining characters are filled at random, so that each data block contains exactly 64 characters.

The size of the key is 64 decimal digits where each decimal digit takes values from 0 to 7. Hence 64^8 different keys are possible. It is estimated that on a 4GHz single core processor the time required to encipher/ decipher a text block is 18μsec. Hence, the vulnerability to brute force attack is very less [the life time of a human being i.e., 100years is approximately equal to 3Gsec, the time required to try all possible keys to decipher a single cipher block by brute force method is roughly 5Gsec]

It is estimated that to encipher a text book containing 500 pages, each page having 40 lines and each lines having 40 characters the time required is 7½minutes.

1. O. Acricmez, C.K. Koc, J.P. Seifert "Predicting secret keys via branch prediction" in Proc. RSA(CT-RSA)2007, Lecture Notes in Computer Sciences, Vol.4377 (Springer Berline, 2007) pp.225-242.
2. Adams C. "Simple and Effective key scheduling for symmetric ciphers" proceedings, workshop in selected areas of cryptography SAC'94,1994
3. Anna M. Johnston, Peter S. Gemmell, "Authenticated key exchange Provably Secure against the Man-in-Middle Attack", Journal of Cryptology (2002) Vol. 15 Number 2 pages 139-148.
4. Carlistle Aams and Stafford Tavares, "The Structured Design of Cryptographically good s-boxes, Journal of Cryptology, 1990, Vol.3, No.1, Pages 27-41.
5. Denis X Charles, Kristin E. Lauter and Eyal Z. Goren, "Cryptographic Hash Functions from Expander Graphers", Journal of Cryptology (2009), Vol. 22 Number 1 pages 93-113.
6. Eli Biham, "Cryptanalysis of multiples modes of operation", Journal of Cryptology, 1998, Vol.11, No.1,pgs 45-58.
7. Ivan B. Damgard and Lars R, Kundsen, "Two-key Triple Encryption" Journal of Cryptology (1998), Vol. 11, Number 3, pages 209-218.
8. John Blaack and Phillip Rogaway "CBC MACs for Arbitrary-Length Messages: The Three-key constructions" Journal of cryptology (2005) Vol. 18 pages 111-131.
9. J. Kelesey B. Schneir and D. Wagner, "key-schedule cryptanalysis of IDEA, G-DEA, GOST, SAFER and triple-DES. In N. Koblitz editor", Advances in cryptology-Proc.CRYPTO'96, LNCS 1109, pages 237-251.Springer-Verlag Berlin 1996.
10. Lester Hill, "Concerning certain linear transformation apparatus of cryptography", The American Mathematical monthly, March 1931, pp 135-154.
11. Lorenz Minder and Alistair Sinclair, "The Extended k-tree Algorithm" Journal of cryptologyDOI: 10.1007/s00145-011-9097-y.
12. Minh-Huyen Nguyen and Salil Vadhan, "Simple Session Key-generation from the short random passwords", Journal of Cryptology(2008), Vol. 21,Number 1, pages52-96.
13. Moses Loskov, Ronald L. Rivest and David Wagner, "Tweakable Block Ciphers", Journal of Cryptology (2011), Vol.24, Number 3, pages 588-613.
14. Victor Shoup and Rosario Gennaro "Securing Threshold Cryptosystems against Chosen Cipher text Attack, Journal of Cryptology (2002) Vol15, Number2 pages75-96.
15. Xiaolin Wang, Guoquin Chen, Jianqin Zhou "A title note on linear transformations in cryptography", Journal of Information Engineering and Electronic Commerce 2009, IEEE'09 International symposium 16-17th May 2009.