Modified Version of Playfair Cipher using Linear Feedback Shift Register and Transpose Matrix Concept

Vinod Kumar, Santosh kr Upadhyay, Satyam Kishore Mishra, Devesh Singh

***Abstract*—In this paper we are presenting a new technique for secure transmission of message by modified version of playfair cipher combining with random number generator and transpose of matrix concept. To develop such method of encryption technique we have used one of the simplest methods of random number generator called Linear Feedback Shift Register and Transpose Matrix concept has been used. The previous playfair cipher method is based on polyalphabetic cipher which is relatively easy to break because it leaves much of loop hole and a small hundreds of letters of cipher text are sufficient. Here we are generating random number sequences and placing it into 6X6 matrix. Then finding the transpose of it and mapping it to secret key of playfair cipher method. Corresponding number s will be transmitted to the receiver instead of alphabetic numeric key. This method increases security of the transmitted key over unsecured transmission media.**

Index Terms*— *Random number, Playfair Cipher, Poly-alphabetic-Numeric cipher, Linear Feedback Shift Register.

I. INTRODUCTION

In Cryptography,[1] the generation of random numbers are used very heavily. So there is a logical circuit called Linear Feedback Shift Register (LFSR)[2] which is good circuit for generating various random number sequences. We can easily modify LFSR and generate different random number sequence. Due to this it give good security for transmission of secret key. The hardware and software implementation of LFSR is also easy due to this it can be used very efficiently for the generation of random number sequence. This paper presents an approach with LFSR with transpose matrix concept and Playfair cipher. In this paper, the numbers and alphabets are arranged 6X6 table based on secret key. Due to large number of combination of alphabets and number (26! X 10! = 10^{33}) it becomes very difficult to break.

Revised Manuscript Received on June 06, 2013.

Vinod Kumar, Department of Computer Science and Engineering, Mewar
University, Chittorgarh(Rajasthan), India

Santosh Kumar Upadhyay, Department of Computer Science and Engineering, Mewar University, Chittorgarh(Rajasthan), India,

Sattyam Kishor Mishra, Department of Computer Science & Engineering, Meerut Institute of Engineering and Technology, Meerut(Uttar Pradesh), India

Davesh Singh, Department of Computer Science and Engineering, Meerut
Institute of Engineering and Technology, Meerut(Uttar Pradesh), India

II. THE PLAYFAIR CIPHER

In the 18th century, it was first invented by Charles Wheatstone but it has mainly used by Lord Playfair[3]. On that day the given plaintext arranges in the table which is a 5X5 matrix. That table was based on only alphabetic cipher in which I and J placed in one column because 5X5 matrix can store 25 alphabet but we have 26 alphabets as shown in Table 2.1. Now we are using a table which is a 6X6 matrix to arrange the plaintext which is based on alphabetic and numeric cipher [4]. This method arranges the given plaintext in the table which is based on key value. This is described as follows with key shown in table 2.2. Here we are using 26 letters and 10 numbers (0 to 9), so there is 36 X 36 = 1296 diagrams will be produced so it becomes very difficult to find particular structure. In this we are filling table character followed by digits after filling the key in table. First we fill remaining character then followed by remaining number.

Key: KEYWORD

K	E	Y	W	O
R	D	A	B	C
F	G	H	I/J	L
M	N	P	Q	S
T	U	V	X	Z

Table 2.1 Plaintext Based on Only Alphabet

The above table show that the arrangement of 26 alphabets into 5X5 matrix which has used by Lord Playfair in which I/J is placed in same box as we have space for only 25 alphabets. These two alphabets very rarely used alphabets.

The keyword (combination of alphabet only) provided by the user that must be converted to upper case . We could have chosen to use small letters for alphabet in the keyword; the goal is to have some uniform convention. Now to make the matrix table, we have taken a keywords (combination of alphabet only) that must be converted to upper case and the result of removing the keywords from the alphabet. Finally the combined word must be rearranged into 5X5 square matrix such that first we fill keyword then we fills remaining characters as shown in data flow diagram Fig 2.1.

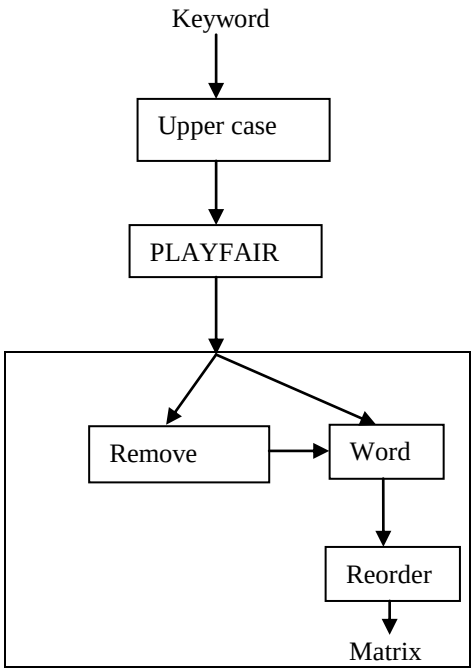

Fig. 2.1 Data Flow Diagram for Existing Model

Key: PLAY34FI5R

P	L	A	Y	3	4
F	I	5	R	B	C
D	E	G	H	J	K
M	N	O	Q	S	T
U	V	W	X	Z	0
1	2	6	7	8	9

Table 2.2 Proposed Plaintext Based on Alphabet and Digits

The above table 2.2 show that the arrangement of 26 alphabets and 10 digits into 6X6 matrix which has used in our paper in which I/J is placed in different box as we have space for each alphabets and digits. This paper presents an approach with LFSR with transpose matrix concept and Playfair cipher. In this paper, the numbers and alphabets are arranged 6X6 table based on secret key. Due to large number of combination of alphabets and number (26! X 10! = 10^{33}) it becomes very difficult to break.

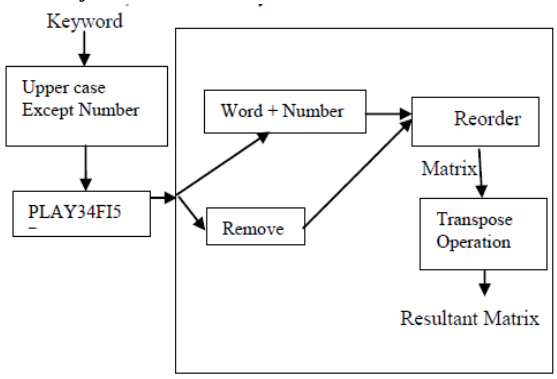

Fig. 2.2 Data Flow Diagram for Proposed Model

The keyword (combination of alphabet and number only) provided by the user that must be converted to upper case except number. We could have chosen to use small letters for alphabet in the keyword; the goal is to have some uniform convention. Now to make the matrix table, we combine words and numbers with keyword and the result of removing the keywords from the alphabet and numbers. Finally the combined word must be rearranged into 6X6 square matrix such that first we fill keyword then we fills remaining characters followed by remaining number. After rearranging the matrix we apply transpose operation on the matrix called resultant matrix as shown in data flow diagram Fig 2.2. Now the resultant matrix is used for the mapping purpose.

III. LINEAR FEEDBACK SHIFT REGISTER

A linear feedback Shift Register is a simply shift register with input is a linear function of its output and previous state. In this the output of some flip-flop is XOR-ed with the output of other which helps to produce the various random sequences. In this we need to choose an initial value of Shift Register to get various random sequences which is called seed. The position of the bit that affects the next state is called tap which allow producing random sequences by varying it.

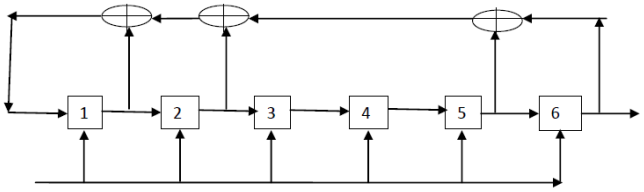

Fig. 3.1 Feedback connection [6, 5, 2, 1].LFSR of length m=6

In the above circuit shown in fig.3.1, a **flip-flop and X-OR** device is used. In this circuit state of the flip-flop is shifted to the next state at each pulse which is down to the line. It also computes the Boolean function of each state of the flip-flops. As the sequence generated by LFSR with m-flip-flop that cannot exceed 2^m-1 due to this that sequence is called m-sequence.

Feedback Symbol	State of Shift Register						Output Symbol
1	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
1	1	1	1	0	1	1	0
0	1	1	1	1	0	1	1
1	0	1	1	1	1	0	1
0	1	0	1	1	1	1	0
1	0	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	1	0	1	0	1	1
1	1	1	1	0	1	0	1
1	1	1	1	1	0	1	0
1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	0	0	1	1	1	1	1
After 63 iteration							
0	1	0	0	0	0	0	0

Table 3.1 Maximal Lengths Tap Sequence Produced by the LFSR

LFSR can have more than one maximal length tap sequences. A LFSR which generates a maximal length tap sequences also describes the exponents which is known as polynomial mod 2. For example, a code sequence of table 3.1 describes the polynomial **x^5+x^1+1**.If we modify the taps and logical function in the LFSR circuit, produce the various random sequences.

Code: 000001101111.....

The code sequence of table 3.1 comes to initial state after 63 iteration because here we have taken m=6 which produces 2^m-1 code sequences, which is similar to 2^6-1.

IV. PURPOSED ALGORITHM

Algorithm1:

I/P: Any Keyword(combination of alphabets and number), seed (initial input given to LFSR), matrix table of size 6X6 of proposed plaintext

O/P: Encrypted Keyword which has to transmitted on transmission media

1. While(seed != next 6 output grouped bits)
2. for each out bit generated by LFSR group in to 6 bits and store into an array A[M]
3. For i = 0 to M
4. Convert(A) // Binary to decimal(random sequences)
5. M'=TRANSPOSE(M)
6. Map M' to matrix table of size 6X6 of proposed plaintext

Algorithm2:

Convert (A)

1. num, bnum, dec = 0, base = 1, rem, bnum = *A ;
2. For i = 0 to M
3. {
4. Num=A[i]
5. while(num > 0)
6. {
7. rem = num % 10;
8. dec = dec + rem * base;
9. num = num / 10 ;
10. base = base * 2;
11. }
12. each random sequences(dec) placed into 6X6 matrix M using any rules of classical playfair cipher
13. }

The new algorithm having many advantages over the basic playfair cipher which has been implemented. In this algorithm, we are arranging the random sequences generated by LFSR(by grouping 6 bits at a time) are arranged in the table which is of size 6X6 using any rules of classical playfair cipher . After filling up the table, the algorithm will find the transpose of the matrix of size 6X6 which is filled by random sequences. Now the resulting matrix will be mapped to plaintext in the table. The output code bits of the above LFSR is 000001 101111 010111 111100 101100 011110, m=6.By increasing the number of flip-flops, cycle length will get increased. Here the output code bit are grouped as follows, here 6 bits are grouped 000001, 101111, 010111, 111100, 101100, 011110. The grouped bits having equivalent number are filled in the table using any rules of classical playfair cipher.

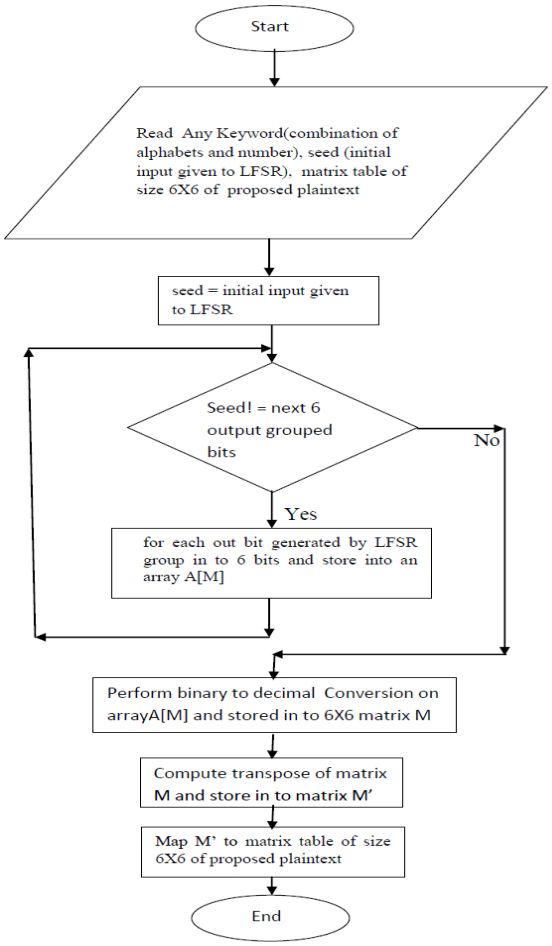

Fig. 4.1Flow Chart for Proposed Algorithm

60	23	47	44	23	30
47	1	44	23	60	22
44	28	60	1	22	47
23	44	47	23	30	60
60	30	28	22	44	1
1	60	30	28	47	43

Table 4.1Arrangement of Code Sequence from LFSR

Here we have taken m=6, the initial state of LFSR is 10000 taken, so LFSR circuit will generate up to 2^m-1 sequence. In this the generator returns to the initial state 100000 after 63 iteration. Now the plaintext filled in the matrix which is based on the key value. After filling up the table we find the transpose of the matrix which helps to increase one more level on security [5, 6].

60	47	44	23	60	1
23	1	28	44	30	60
47	44	60	47	28	30
44	23	1	23	22	28
23	60	22	30	44	47
30	22	47	60	1	43

Table 4.2 Arrangement of Code Sequence after Transpose of Matrix Table 4.1

The above table 4.2 shows the arrangement of code sequence which comes after taking the transpose of table 4.1 .The table 4.1 shows the arrangement of code sequence which is arranged according to the sequence generated by LFSR circuit [7].

60	47	44	23	60	1
23	1	28	44	30	60
47	44	60	47	28	30
44	23	1	23	22	28
23	60	22	30	44	47
30	22	47	60	1	43

P	L	A	Y	3	4
F	I	5	R	B	C
D	E	G	H	J	K
M	N	O	Q	S	T
U	V	W	X	Z	0
1	2	6	7	8	9

Table 4.3 Mapping of Code Sequence after Transpose of Matrix to Plaintext

Now the values of alphabets (A-Z) and number (0-9) as follows:

{P, L, A, Y, 3, 4} = {60, 47, 44, 23, 60, 1}
{F, I, 5, R, B, C} = {23, 1, 28, 44, 30, 60}
{D, E, G, H, J, K} = {47, 44, 60, 47, 28, 30}
{M, N, O, Q, S, T} = {44, 23, 1, 23, 22, 28}
{U, V, W, X, Z, 0} = {23, 60, 22, 30, 44, 47}
{1, 2, 6, 7, 8, 9} = {30, 22, 47, 60, 1, 43}

Now we are providing the encryption [8] based on classical playfair cipher. The proposed method follows the same rules and regulations.

Example:

Plaintext: PAN23DIT

PA → LY, N2 → VL, 3D → PJ, IT → CN

Cipher text:

{(L, Y), (V, L), (P, J), (C, N)}= {(47, 23), (60, 47), (60, 28),(60, 23)}

Now the transmitted cipher text is {(47, 23), (60, 47), (60, 28), (60, 23)} instead of combination of alphabetic and numeric. Decryption process is the inverse of encryption process.

V. ANALYSIS OF PROPOSED METHOD

This proposed method increases the security of the cipher text and also the inner structure of this method is very simple. Currently numbers of algorithms are available for encryption but it requires many complex rounds like DES, AES etc. AES and DES which uses two concepts for security, encryption and decryption. Encryption means relationship between plaintext and cipher text as complex as possible. Decryption means mask the statistical properties of data in the cipher text. Our approach allows encryption and decryption [9] can be easily incorporated to Playfair Cipher. In this we have used a LFSR circuit which can be used to generate random sequences. Unpredictable different random sequences can be produced from LFSR by varying logic functions and taps. Increasing the number flip-flops, can increases the cycle length. It can be easily implemented with advent of new computer. The implementation of LFSR in hardware and Software is very easy as compared to other. It is very cost effective and also speed is considerably very high compare to other methods. This method of encryption does not increase size of the cipher text [10]. In the areas which having low bandwidth or less memory storage this method can be used efficiently. The classical Playfair cipher is relatively easy to break because it still leaves much of the holes of the plaintext language. This method of incorporating random sequences can also be applied to other ciphers.

VI. CONCLUSION

This paper has attempted to implement modified Playfair cipher using Linear Feedback Shift Register and transpose of matrix concept. The classical Playfair cipher is not to much secure because it produces only 676 structures. By filling the random sequences into 6X6 matrix and then taking the transpose of that matrix. After this mapping of resulting matrix to classical Playfair cipher, increases the security of the message transmission by many folds. Another approach to further enhance the security is to use high-dimensional chaotic or hyperchaotic systems such as the Rössler or the Lorenz systems.

REFERENCES

1. Schnier B. “Applied cryptography: protocols, algorithms and source code in C”. New York: John Wiley and sons; 1996.
2. http://en.wikipedia.org/wiki/Linear_feedback_shift_register.html.
3. Johannes A.Buchmann Introduction to Cryptography,Second Edition 2001, Springer – Verlag NY, LLC
4. Behrouz A. Forouzan. “Cryptography and Network Security”, Special Indian Edition 2007, The McGraw- Hill companies, New Delhi
5. Dhiren R.Patel “Information Security Theory and Practice” First Edition,2008, Prentice-Hall of India Private Limited.
6. Keith Harrison, Bill Munro and Tim Spiller “Security through uncertainty “HP Laboratories, February 2007.
7. Simon Haykin , “ Communication Systems ,” 4th Edition , Willey.
8. William Stallings, “ Cryptography and Network Security Principles and Practice “ Second edition,Pearson Education.
9. Wayne Tomasi “Electronic Communications System Fundamentals through Advanced,” 5th edition,Pearson Education, 2008.
10. Menezes AJ,Oorschot PCV, “Vanstone SA Handbook of applied cryptography” . Boca Raton , Florida , USA : CRC Press ; 1997.

AUTHORS PROFILE

Vinod Kumar: Born in November 1984 in Gaya(Bihar). Phone number: +919368269427 & E-mail id: vinod4r@gmail.com. He had done B.E (Computer Science & Engg.) from Bhagalpur college of Engineering(T.M.B. University). M.Tech (Computer Science & Engineering) From Mewar University Chittorgarh (Rajasthan**).**He has 3 year Teaching Experience from Various Engineering Institute before his M.Tech . The major field of Study Area is Automaton Theory, Compiler Constructiion,Algorithm, Programming and Operating System.

Santosh Kumar Upadhyay:Birth Place & Date - Azamgarh (U.P.), INDIA on 05/08/1982. Master of Technology from Tezpur University(A central University), Tezpur (Assam), INDIA. He is silver medalist in Master of Technology in Tezpur University. The major fields of study is network security, data mining,Algorithm. He has more than seven years experience in teaching and research as ASSISTENT PROFESSOR. He is working at Galgotias College of Engineering & Technology , Greater Noida (U.P.), INDIA. He has published many papers in international Journals .

Sattyam Kishor Mishra:Birth Place & Date -Nainital(Uttrakhand), INDIA on 31/12/1984.Master of Technology from Maulana Azad National Institute of Technology, Bhopal, INDIA.The major fields of study is information security,network security, computer security, Algorithm.He has more than three years experience in teaching and research as ASSISTANT PROFESSOR. He is working at Meerut Institute of Engineering and Technology, Meerut (UP), India.

Davesh Singh Som received his Bachelor.s degree in Computer Science & Engineering in 2007 from Radha Govind Enginering College, Meerut, India. Presently, he is at the verge of completion of M.Tech in Computer Science&Engineering fromMeerut Institute of Engineering and Technology, Meerut, Uttar Pradesh, India. His research areas are Computer Network, Data Structure, Computer Organization and Object Oriented Techniques.