11 × 11 Playfair Cipher based on a Cascade of LFSRs

Ouday Nidhal Ameen Hanosh[1], BaraaWasfi Salim[2]

[1] *(Computer and Communication Eng. Department, College of Computer and IT, Nawroz University, Iraq)*
[2] *(Computer Science Department, College of Computer and IT, Nawroz University, Iraq)*

Abstract: *Playfair cipher is one of the better-known multiple letter encryption ciphers. In this method, the diagrams in the plaintext are treated as a single unit and then these units converted into ciphertext diagrams. This paper implements and discusses a new system which proposed by using 11 × 11 Playfair cipher that supports all 26 alphabets in both: upper case letters (A-Z) as well as lower case letters (a-z), ten digits (0-9), special characters and the extended special characters. This combination will tackle the limitation of 5×5 Playfair cipher in which both "i" and "j" letters could not appear simultaneously. In order to increase the level of security of this method, the output of an 11×11 Playfair procedure will be an input to the complete procedure of a cascade LFSRs. Finally, this system was implemented using MATLAB 8.0 (R2012b).*

Keywords: *Playfair Cipher, Secret Keyword, Special Characters, LFSRs, Cryptanalysis.*

I. Introduction

Playfair cipher is a primitive block cipher [1]. It is a famous biographic or Polygraphic substitution cipher which encipher block of letters, actually two letters at a time instead of a single letter. This leads that the cryptanalysis will become harder, as it destroys the single letter frequency distribution. The cryptanalysis of the Playfair cipher is aided by the fact that the single unit diagram and its reverse will encrypt in a similar fashion. That is, if AB encrypts to XY, the BA will encrypt to YX [2] [3]. So by discovering words that start and end in reversed diagram it is easy to compare them with plaintext word that are similar. This leads to modify the Playfair cipher with LFSR for random number generator.

This paper presents a new approach for encipher which uses 11×11 Playfair matrix based on secret keywords along with a cascade of LFSRs. Furthermore, the characters of the plaintext in this paper belong to the set of ASCII characters denoted by the codes 0 to 255. This will enhance the cipher significantly and decreases the possibility of cryptanalytic attack. For this an analysis of all pitfalls and security loopholes will be taken into consideration in order to provide a new cipher which is strong enough.

The section 2 of this paper deals with the work overview which gives a full description of 11×11 Playfair matrix. The new proposed system for encryption and decryption is described in section 3. Section 4 presents a practical example of our system. Section 5 involves with cryptanalysis. Finally, conclusion is provided in section6.

II. Work Overview

Recently, the Playfair cipher was extended by using 6×6 or 8×8 matrix, so it would be using 36 and 64 grids respectively [4] [6]. In our system not only the alphabet in both cases (upper and lower) encrypts but also the numerals, special characters and an extended special characters. At encryption time, the symbol ■is being used to provide space between two words. Moreover, # is used as filler character in order to separate two alphabets if they are repeated in pair. # will also be used to put at the end if the number of plaintext characters is an odd number. During the decryption time,■ will be substituted by blank space of one alphabet while, the # symbol will be suppressed.

TABLE1: 11×11 Playfair matrix based on secret key {NAWROZ Now @Duhok!}

{	N	A	W	R	O	Z	o	w	@	D
u	h	k	!	}	B	C	E	F	G	H
I	J	K	L	M	P	Q	S	T	U	V
X	Y	a	b	c	d	e	f	g	i	j
l	m	n	p	q	r	s	t	v	x	y
z	0	1	2	3	4	5	6	7	8	9
#	$	%	^	&	*	(	)	-	=	+
[	]	;	‘	:	“	\	,	.	/	<
>	?	£	¥	α	β	π	σ	μ	τ	∞
±	≥	≤	÷	Ç	â	ä	à	å	ç	ê
ë	è	ï	î	ì	Ä	Å	É	æ	Æ	■

Table 1 is used for encryption and decryption procedure by taking into account the secret keyword. The attacker now needs to search in 121×121 = 14641 diagrams rather than 64×64, 36×36, or even 26×26 diagrams. This definitely increases the resistance dramatically against brute force attack.

1.1 PROBABILITY OF OCCURRENCE

Table 2 shows the probability of occurrence of different modified Playfair ciphers. The value of occurrence of our work is far less when compared, and frequency analysis is became now a tough issue.

TABLE2: Probability of Occurrence of different modified Playfair Ciphers.

Modified Playfair ciphers	Probability of Occurrence
5×5	0.0384
6×6	0.0277
8×8	0.0156
11×11	**0.00826**

III. Proposed System

The first step in our proposed system is to convert the plaintext into a matrix of size 2×n, where n represents the length of the plaintext.

N = length (pt)
pt = pt (i, j) ; i = 1: n ; j = 1: 2

$$pt = \begin{bmatrix} pt_{11} & pt_{12} \\ pt_{21} & pt_{22} \\ . & . \\ . & . \\ . & . \\ pt_{n1} & pt_{n2} \end{bmatrix}$$

The next step involves constructing and using an 11×11 arranged matrix and the secret keyword after suppressing the repeated letters. After that the plaintext matrix (pt) can be converted into corresponding Ciphertext matrix (Ct^*) by applying the following three rules:

1) If the two characters appear on the same row of an 11 × 11 Playfair matrix, then both of them can be replaced with the characters to their immediate right respectively.
2) If the two characters appear on the same column of an 11 × 11 Playfair matrix, then both of them can be replaced with the characters immediately below respectively.
3) If both of the characters didn't lies on the same row or column, then replace them with the characters on the same row respectively but at the column of the other characters of pair.

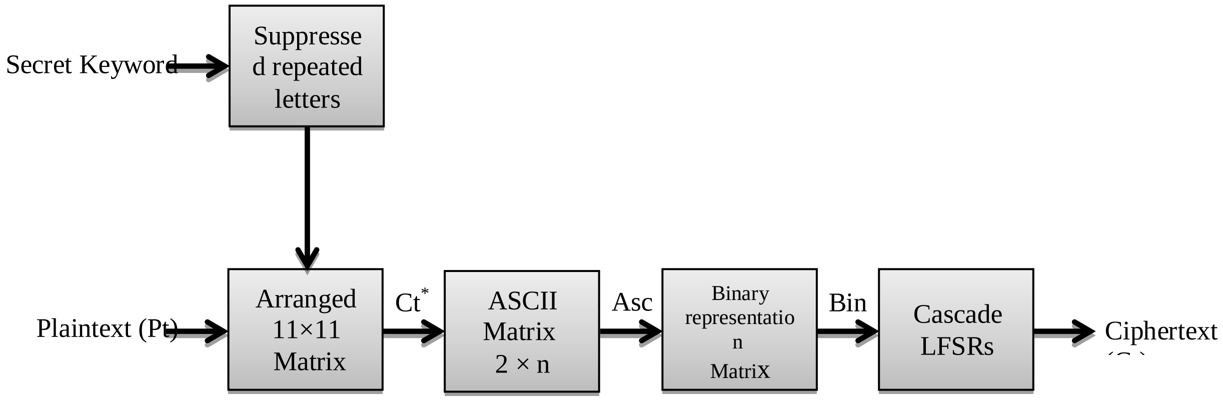

Fig. 1: Block diagram explaining the encryption proposed system.

The size of the resultant ciphertext matrix (Ct^*) must be equal to the size of the plaintext matrix (pt).

$$Ct^* = \begin{bmatrix} Ct_{11}^* & Ct_{12}^* \\ Ct_{21}^* & Ct_{22}^* \\ . & . \\ . & . \\ . & . \\ Ct_{n1}^* & Ct_{n2}^* \end{bmatrix}$$

The elements of the Ct^* matrix are then converted to their corresponding decimal ASCII values. The resultant matrix called Asc.

$$Asc = \begin{bmatrix} Asc_{11} & Asc_{12} \\ Asc_{21} & Asc_{22} \\ . & . \\ . & . \\ . & . \\ Asc_{n1} & Asc_{n2} \end{bmatrix}$$

Then the elements of Asc matrix are converted into their corresponding binary values. Each element consists of 8 bits as ASCII values range from 0 to 255. This matrix called Bin.

$$Bin = \begin{bmatrix} Bin_{11} & Bin_{12} & . & . & . & Bin_{116} \\ Bin_{21} & Bin_{22} & . & . & . & Bin_{216} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ Bin_{n1} & Bin_{n2} & . & . & . & Bin_{n16} \end{bmatrix}$$

Now, the binary representation matrix Bin which represents the output of the Playfair encryption algorithm will be applied as an input to a cascade of LFSR, in order to get the permuted sequence of bits. LFSR is a shift register of length L which consists of L delay elements numbered 0,1, ….., L-1 each stage capable of storing one bit and having one input and one output; and a clock which controls the movement of data. The input state of LFSR is a linear function of its previous state. The only linear functions of single bit are XOR and inverse-XOR, thus it is shift register whose input bit is driven by the exclusive or of some bits of the overall shift register value [5] [7]. Figure 2 shows a cascade of LFSR which is used in our design.

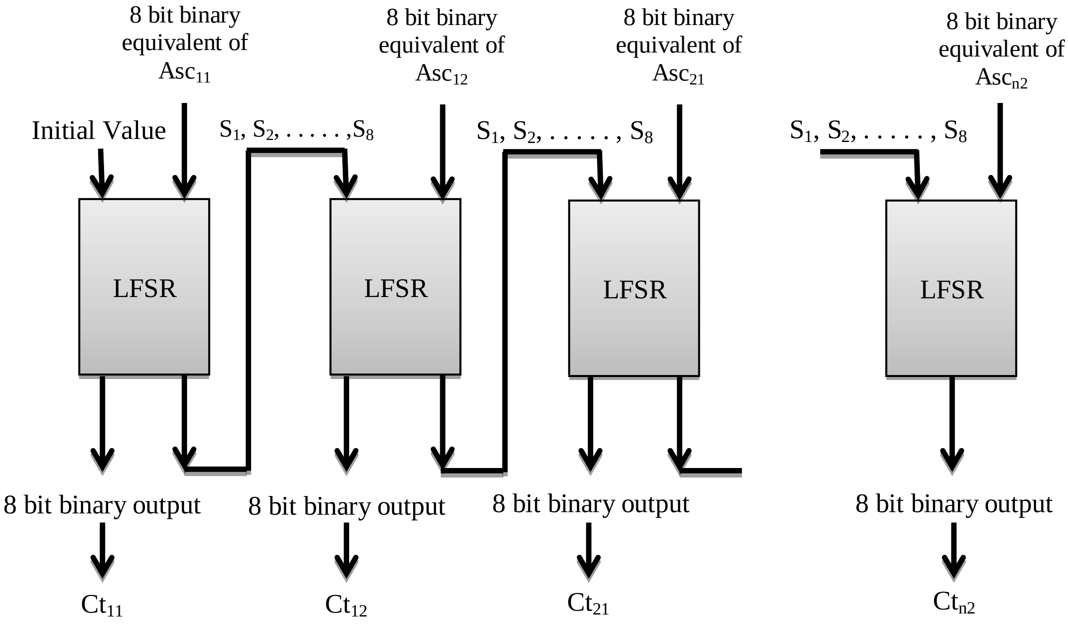

Fig. 2: A cascade of LFSRs.

It is important to choose the design of the LFSR because not all the feedback polynomials are adequate to use in cryptography. In addition to that the choose of initial value is also important cause it acts as a secondary key in the cipher. Any change in the initial value will affect the overall output sequence. The designer is the only person who knows the design of the LFSR and also the initial value. The following figure illustrates an 8 bit LFSR that used in our design.

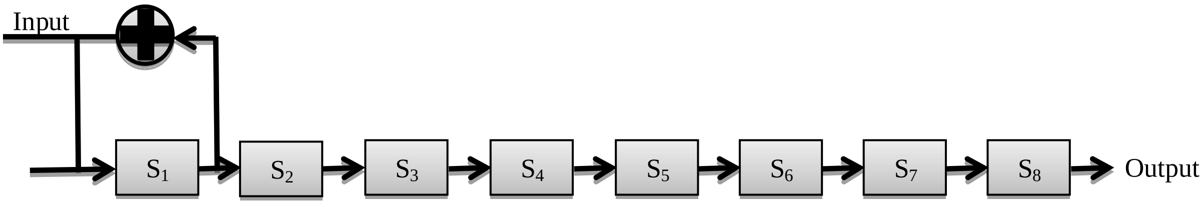

Fig. 3: An 8-bit LFSRs.

A cascade of LFSR procedure:

One of the most important things in the design of a cascaded LFSR is that the last row of the first LFSR will be the initial value to the next LFSR and so on; this will lead to increase the security. Because we used 8-bit LFSR in our design so we named each bit in LFSR as S_1,S_2S_3,S_4,S_5,S_6,S_7,S_8, the equivalent binary representation for all the elements of matrix Asc will be taken as an input to the LFSR. This input is called X where

X= {X_1,$X_{2,}X_3$,X_4,X_5,X_6,X_7,X_8} = binary equivalent for every element of Asc matrix.

Since we shifted one bit to the right at a time and the number of output of each LFSR is fixed. So we can store the sequence of output for each 8bit input into Y where

Y = {Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8}

The procedure for each binary input will goes like

```
Set i = 1
While ( i<= 8)
{
Y[9-i] = S[8]
SHR(S[i]) /* shift the bits in S to the right
S[1] = S[1] xor X[9-i]
i=i+1
}
```

Note that, this procedure will be repeated for every 8 bit cascaded LFSR in our design. After we finished the LFSR procedure we get the corresponding 8 bit permuted binary sequence for the 8 bit binary input. These binary sequences will converted to decimal to get the last Cipher text which represented as a matrix Ct.

$$Ct = \begin{bmatrix} Ct_{11} & Ct_{12} \\ Ct_{21} & Ct_{22} \\ . & . \\ . & . \\ . & . \\ Ct_{n1} & Ct_{n2} \end{bmatrix}$$

IV. Cipher Presentation

Let choose the following Plaintext.
“ Even the stopped clock is right twice a day. “
The first step is to replace the blank space with the ■symbol.
So, the plaintext becomes
“■ Even ■ the ■ stopped ■ clock ■ is ■ right ■ twice ■ a ■ day. ■ “
For simple presentation we have concentrate on the first 18 characters of the plaintext, i.e.
“■ Even ■ the ■ stopped ■ clo
The 18 characters of the plaintext is now arranged into 9 × 2 matrix form

$$Pt = \begin{bmatrix} '' & E \\ v & e \\ n & t \\ h & e \\ s & t \\ o & p \\ p & e \\ d & c \\ l & o \end{bmatrix}$$

By using Table1 that mentioned in section tow and the secret keyword {NAWROZ Now @Duhok!}, we can encrypt the plaintext in order to get Ct*

$$Ct^* = \begin{bmatrix} , & B \\ Q & j \\ p & v \\ C & Y \\ t & v \\ W & t \\ s & b \\ e & d \\ t & \{ \end{bmatrix}$$

Each element in the Ct* matrix will converted into an equivalent ASCII code

$$Asc= \begin{bmatrix} 44 & 66 \\ 81 & 106 \\ 80 & 118 \\ 67 & 85 \\ 116 & 118 \\ 87 & 116 \\ 115 & 98 \\ 101 & 100 \\ 116 & 123 \end{bmatrix}$$

By replacing the elements of Asc with their equivalent binary representation we get the matrix Bin

$$Bin= \begin{bmatrix} 0010110001000010 \\ 0101000101101010 \\ 0101000001110110 \\ 0100001101010101 \\ 0111010001110110 \\ 0101011101110100 \\ 0111001101100010 \\ 0110010101100100 \\ 0111010001111011 \end{bmatrix}$$

The values of Bin are entered to the cascade of LFSRs; we presented the first two steps of a cascaded LFSR

I/P = 00101100 Initial Value = 10101010 O/P = 01010101 = 85

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	O/P
1	0	1	0	1	0	1	0	-
1	1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	0	1
0	1	1	1	0	1	0	1	0
1	0	1	1	1	0	1	0	1
1	1	0	1	1	1	0	1	0
0	1	1	0	1	1	1	0	1
0	0	1	1	0	1	1	1	0
0	0	0	1	1	0	1	1	1

I/P = 01000010 O/P = 11011000 =216

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	O/P
0	0	0	1	1	0	1	1	-
0	0	0	0	1	1	0	1	1
1	0	0	0	0	1	1	0	1
1	1	0	0	0	0	1	1	0
1	1	1	0	0	0	0	1	1
1	1	1	1	0	0	0	0	1
1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0

Bin* represents the binary outputs of LFSR

$$\text{Bin}^* = \begin{bmatrix} 0101010111011000 \\ 0111110011110011 \\ 1001101111110011 \\ 1010101101111100 \\ 1100110000110100 \\ 0100101101001100 \\ 0011010010001011 \\ 1000010011000100 \\ 0011101111001011 \end{bmatrix}$$

Converting this matrix element into equivalent decimal we get

$$\text{Asc}^* = \begin{bmatrix} 85 & 216 \\ 124 & 243 \\ 155 & 243 \\ 171 & 124 \\ 204 & 52 \\ 75 & 76 \\ 52 & 139 \\ 132 & 196 \\ 59 & 203 \end{bmatrix} \qquad \text{Ct} = \begin{bmatrix} U & \text{╪} \\ | & \leq \\ \text{¢} & \leq \\ \text{½} & | \\ \text{╠} & 4 \\ K & L \\ 4 & \text{ï} \\ \text{ä} & \text{─} \\ ; & \text{╦} \end{bmatrix}$$

Where Ct represents the final ciphertext. The decryption process is merely the reverse procedure of the encryption and we can get the plaintext easily.

V. Cryptanalysis

Many methods are used to systematically recover plaintext from ciphertext or even to deduce the encryption key, these methods are: 1) Ciphertext only attack which is known also Brute Force attack, 2) known-Plaintext attack, 3) Chosen-Plaintext attack, 4) Adaptive chosen-Plaintext attack, 5) Chosen-Ciphertext attack, 6) Adaptive chosen-Ciphertext attack.

Regarding to the example depicted in section 4 of this paper, the length of the ciphertext in is 18 × 8 = 144 binary bits and this is equal to the length of the plaintext. So the size of the plaintext space which is to be searched by the attacker is $2^{144} \approx 10^{43.3}$. This will cost the attacker an enormous large time. In this case Brute Force could not be used. Not only the combination of alphabets, numerals, special characters and the extended special characters that used in 11 × 11 extended Playfair cipher but also the unpredictable random sequences that produced from cascaded LFSRs will lead to impossible correlation between Plaintext and Ciphertext. So a Known-Plaintext attack is ruled out.

Since our design permit an easilyincorporation of confusion and diffusion to Playfair cipher and a randomization through LFSR, we can conclude that there is no combination of the Plaintext and the Ciphertext could help the cryptanalyst to break this cipher system.

VI. Conclusion

A new system was presented in this paper based on 11 × 11 Playfair cipher and a cascade of LFSRs. This system rapidly increases the security of ciphertext by going for 11 × 11 extended matrix with an ASCII range from 0 to 255 to tackle the issue of the traditional Playfair cipher, and then a cascade of LFSRs is used to increase the security of the transmission through the use of random numbers concept. Moreover, it's quite easy and cost effective to design and implement it on hardware and software. Our system was built by using MATLAB 8.0 (R2012b).

Regarding to the discussion and analysis made, we can conclude that the proposed system is very effective for area with a very less memory storage and a low bandwidth. It is a very rigid system and it cannot be broken by any attacker.

References

[1] V.K. Pachghare, Cryptography and information Security, PHI Learning, New Delhi, 2010.

[2] William Stallings, Cryptography and Network Security: Principles and Practice. 5th edition, Prentice Hall, January 24, 2010.

[3] Johannes A. Buchmann, Introduction to Cryptography. Second edition, Springer-Verlag NY, LLC, 2001.

[4] Shiv Shakti Srivastava, Nitin Gupta and RajramJaiswal "Modified Version of Playfair Cipher by using 8×8 Matrix and Random Number Generation" in Proceeding of IEEE 3rd International Conference on Computer Modeling and Simulation (ICCMS 2011), Mumbai, pp. 615-617, January,2011.

[5] PackirisamyMurali and GandhidossSnethilkumar, Modified Version of Playfair Cipher Using Linear Feed Back Shift Cipher. International Conference on Information Management and Engineering ICIME, pp. 488-490, 2009.

[6] AmandeepKaur, Harsh Kumar Verma and Ravindra Kumar Singh, 6×6 Playfair Cipher using LFSR based Unique RandomNumber Generator. International journal of computer pplications, Volume 51,no. 2,pp. 30-34, August, 2012.

[7] Dhiren R. Patel, Information Security: Theory and Practice. First edition, Prentice Hall of India Private Limited, 2007.