On adaptive of Generations by using $\mathcal{E}$-$\mathcal{A}$ and $\mathcal{D}$-$\mathcal{A}$ laws

AMEER A.J. AL-SWIDI and Yiezi kadham al-talkany

University of Babylon ,College of Education for pure sciences,

Math. Department.

Waleedabd73@yahoo.com

Abstract

In this paper I will introduce some generations and how encipher and decipher and applying new type of laws (the $\mathcal{E}$-$\mathcal{A}$ and $\mathcal{D}$-$\mathcal{A}$ laws), to adaptive the generations.

Key word: P , C ,k, $\mathcal{E}$-$\mathcal{A}$, $\mathcal{D}$-$\mathcal{A}$, LFSR,FCSR.

1.Introduction:

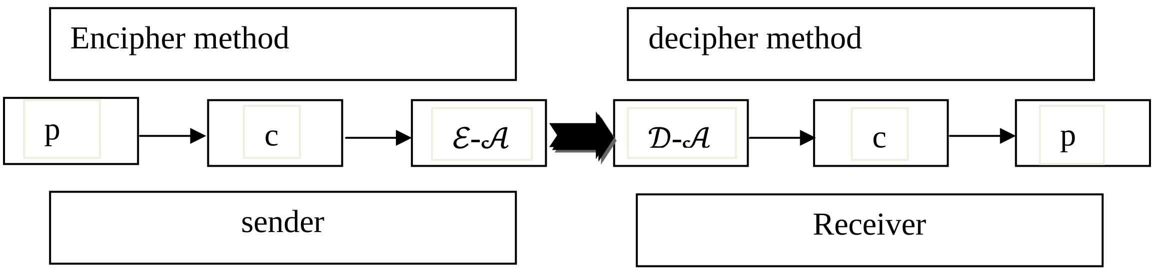
Shift register sequences are used in both cryptography and coding theory ,there is wealthy of theory about them ,stream ciphers based on shift registers have been the workhorse of military cryptography since the beginnings of electronics(cf.[1,2,3,5]) ,

A feedback shift register is made up of two parts: shift register and a feedback function ,the shift registers is a sequences of bits ,(the length of a shift register is figured in bits ,if it is n bits long ,it is called an n-bit shift registers ,each time a bit is needed ,all of the bits in the shift register are shifted 1 bit to the right ,the new left ,most bit is computed as a function of the other bits in the register ,the output of the shift register is 1 bit ,the period of a shift register is the length of the output sequence before it starts repeating cryptographers have liked stream ciphers made up of shift registers: they are easily implemented in digital hardware(cf.[4,6,7]), for adaptive of generations I use $\mathcal{E}$-$\mathcal{A}$ and $\mathcal{D}$-$\mathcal{A}$ laws, where

$\mathcal{E}$-$\mathcal{A}$=min(max(1-c,k),max(c,1-k))≡$\odot$ - operation

$\mathcal{D}$-$\mathcal{A}$=min(max(1- $\mathcal{E}$-$\mathcal{A}$,k),max($\mathcal{E}$-$\mathcal{A}$,1-k))≡ $\circledast$ - operation

And this figure which show as the encipher and decipher ,



Figure(1)

1.1.Linear feedback shift registers:

An n-stage linear feedback shift register(LFSR),consists of a shift register R=(r_n,r_{n-1},….,r_1) and "tap" sequence T=(t_n,t_{n-1},….,t_1),where each r_i and t_i is one binary digit, at each step ,bit r_1 is appended to the key stream ,bits r_n,r_{n-1},….,r_1 are shifted right ,and a new bit derived from T and R is inserted into the left end of the register(see figure(2))

Letting $R'=(r'_n,r'_{n-1},\ldots\ldots,r'_1)$ denoted the next state of R ,we see that the computation of R' is thus:

$$r'_i=r_{i+1} \qquad i=1,\ldots,n-1$$

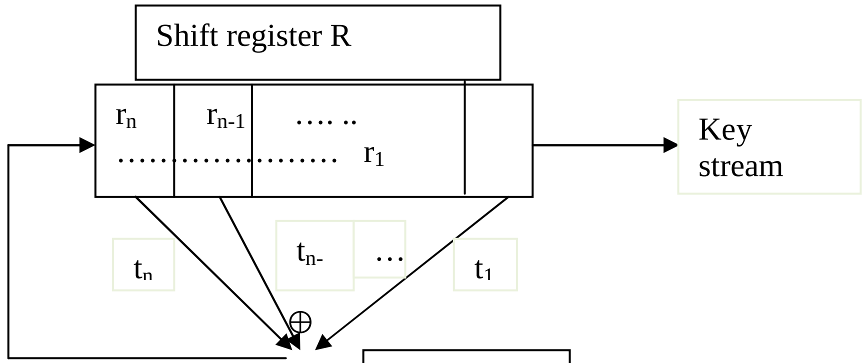
$$r'_n=TR=\sum_{i=1}^{n} t_i r_i \bmod 2$$

$$=t_1r_1\oplus t_2r_2\oplus\ldots\ldots\oplus t_nr_n$$

Thus,

R'=HR mod 2

Where H is the nxn matrix(cf.[2,4]), an n-stage LFSR can generate pseudo-random bit strings with aperiod of 2^n-1.



1.2.Example:

Figure(2)

Illustrates a 4-stage LFSR ,with tap sequence T=(1 0 0 1) ,thus there are "tap" on bits r_1 and r_4 the matrix H is given by:

$$H=\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The polynomial $T(x)=x^4+x+1$ is
primitive ,so the register will cycle through all 15 nonzero bit combinations in GF(2^3) before repeating ,starting R in the initial state 1 0 1 0 ,we have

1 0 1 0

1 1 0 1

0 1 1 0

0 0 1 1

1 0 0 1

0 1 0 0

0 0 1 0

0 0 0 1

1 0 0 0

1 1 0 0

1 1 1 0

1 1 1 1

0 1 1 1

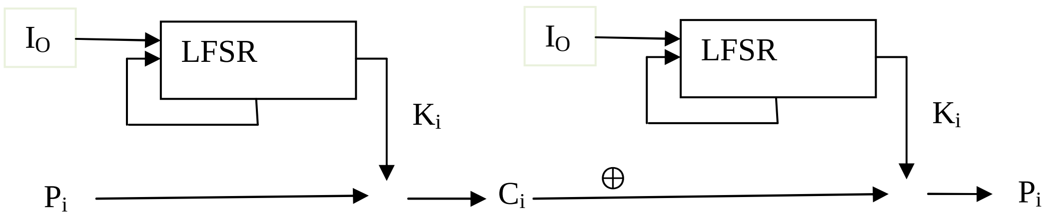
1 0 1 1

0 1 0 1

The right most column gives the key stream ,K=010110010001111….

1.3.Encryption with LFSR:

The binary message stream $P_I=P_1P_2$……, is enciphered by computing $c_i=P_i\oplus k_i$,as the bits of the key stream are generated (see figure(3)) ,deciphering is done in exactly by the same way that is ,by regenerating the key stream and computing $P_i=c_i\oplus k_i$,the seed I_o is used to initialized R for both encipherement and decipherement .



Figure(3)

1.4.Example:

If we take a 4-stage LFSR with tap sequence T=(1001) and initial state 1010,we have

K=010110010001111….

To encipher message M=101010001100111……

$c_i=p_i\oplus k_i$ =111100011101000…….

And ,so the sender is

c_i= 111100011101000…….

And ,to decipher the cipher text

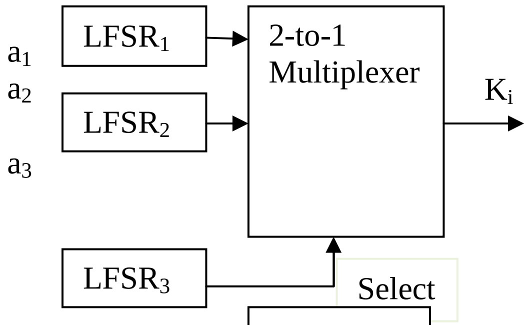
p_i= $c_i\oplus k_i$ =101010001100111…..

1.5.Geffe Generator:

This key stream generator uses three $LFSR_s$,combined in a nonlinear manner(see figure(4)) two of the $LFSR_s$ are inputs into a multiplexer ,and the third LFSR controls the output of the three $LFSR_s$,the output of the Geffe Generator can be described by

$K=(a_1 \wedge a_3)\oplus(a_2 \wedge(-a_3))$

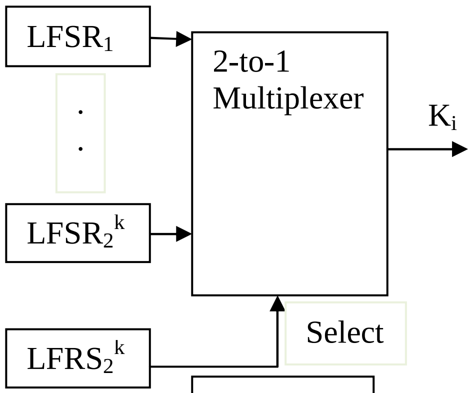
If the $LFSR_s$ have length n_1,n_2 and n_3 respectively ,then the linear complexity of the generator is $(n_3+1)n_2+n_3n_1$,(cf.[1,2,4]).



Figure(4)

1.6.Generalized Geffe Generator:

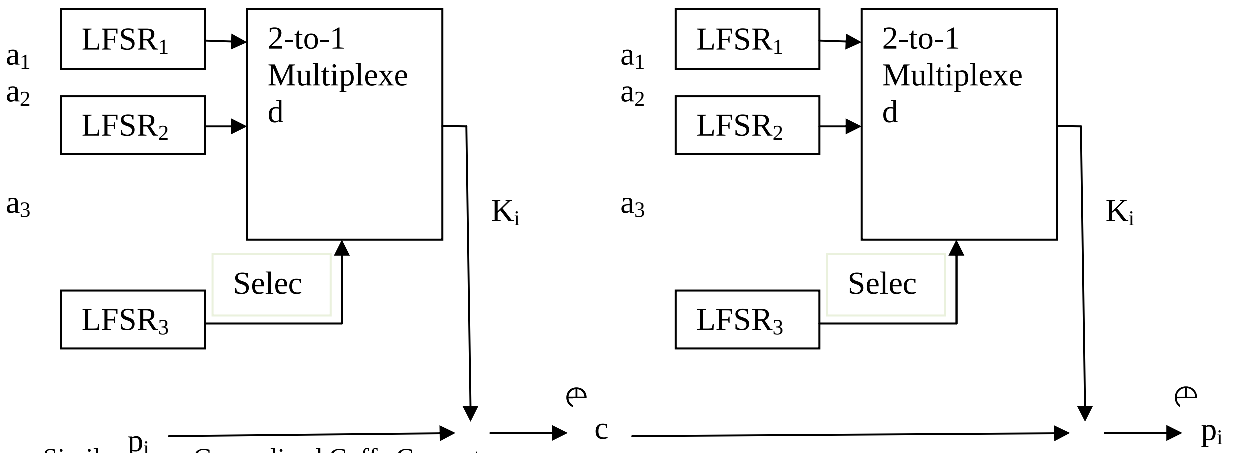
Instead of choosing between two $LFSR_s$,this scheme chooses between K-$LFSR_s$,as long as k is a power of 2,there are k+1 $LFSR_s$ total(see figure(5)) ,$LFSR_{2^k+1}$ must be closed $\log_2^k$ times faster than the other k-$LFSR_s$.



Figure(5)

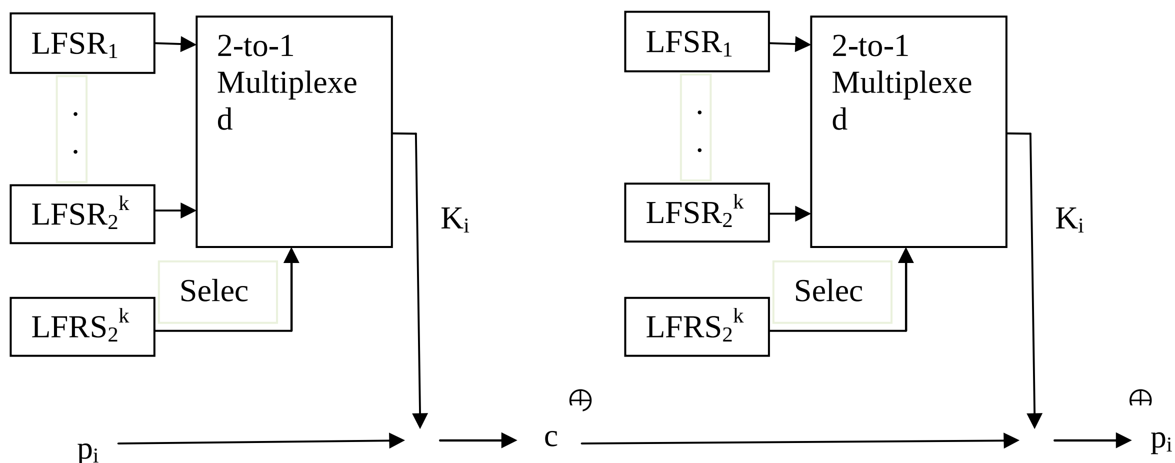
1.7.Encryption with Geffe Generator:

The binary message stream $p=p_1p_2\ldots\ldots$, is enciphered by computing $c_i=p_i\oplus k_i$,as the bits of the key stream are generated (see figure(6)) ,deciphering is done in exactly by the same way that is ,by regenerating the key stream and computing $c_i\oplus k_i=p_i$.



Figure(6)

Similar Generalized Geffe Generator,



Figure(7)

1.8.Example:

If we take a 4-stage $LFSR_1$,$LFSR_2$ and $LFSR_3$ with the same tap sequence T=(1001),with initial state (1101),(0011) and(0101) respectively ,then the key stream is

	a_1=$LFSR_1$	a_2=$LFSR_2$	a_3=$LFSR_3$
1-	1101	0011	0101
2-	0110	1001	1010
3-	0011	0100	1101
4-	1001	0010	0110
5-	0100	0001	0011
6-	0010	1000	1001
7-	0001	1100	0100
8-	1000	1110	0010
9-	1100	1111	0001
10-	1110	0111	1000
11-	1111	1011	1100
12-	0111	0101	1110
13-	1011	1010	1111
14	0101	1101	0111
15-	1010	0110	1011
periodic	1101	0011	0101

Where ,

$K_i=(a_1 \wedge a_3) \oplus (a_2 \wedge (-a_3))$, and $1\wedge1=1$,$1\wedge0=0$,$0\wedge1=0$,$0\wedge0=0$,-1=0,-0=1 ,then

a_1=101100100011110…………

a_2=110010001111010…………

a_3=101011001000111…………

$-a_3$=010100110111000………..

and

K_i=111000000111110…………

Now to encipher the message ,p_i=110111010000110…….

$c_i=p_i \oplus k_i$

=001111010111000………

And decipher ,is

$p_i=c_i \oplus k_i$

=110111010000110…….

1.9.Feedback with carry shift registers:

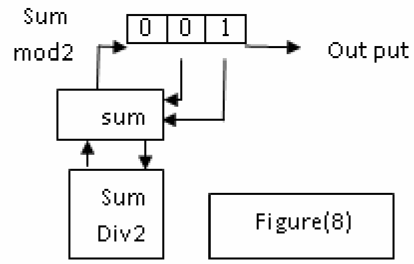
A feedback with carry shift register ,or FCSR is similar to A LFSR ,both have a shift register and a feedback function ,the difference is that a FCSR also has a carry register (see figure(8)), instead of XOR_{ing} all the bits in the tap sequence ,add the bits together and add in the contents of the carry register ,the result mod2 becomes the new bit ,the result divided by 2 becomes the new content of the carry register(cf.[5,6,7]).An example of a 3-bit FCSR tapped at the first and second bit ,its initial state is 001 ,and the initial content of the carry register is 0 ,the output bit is the right most bit of the shift register ,

New bit=(∑tapped position + carry)mod2

New carry=(∑tapped position +carry)div2

Shift register	Carry register
0 0 1	0 q_{12}
1 0 0	0 q_{11}
0 1 0	0 q_{10}
1 0 1	0 q_9
1 1 0	0 q_8
1 1 1	0 q_7
0 1 1	1 q_6
1 0 1	1 q_5
0 1 0	1 q_4
0 0 1	1 q_3
0 0 0	1 q_2
1 0 0	0 q_1

And the length of carry shift register=$\log_2^t$,t-number of tapped position ,and has period of 10 ,(see figure(8))



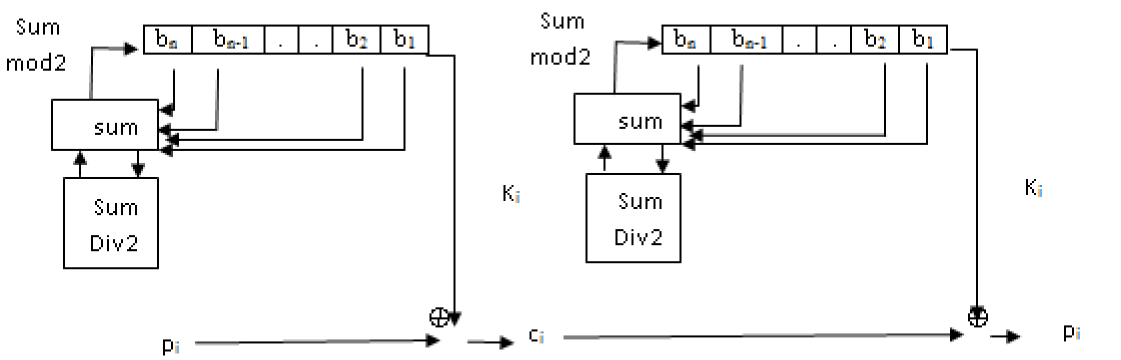
Figure(8)

The maximum period of a FCSR is not 2^n-1 ,where n is the length of the shift register ,the maximum period is q-1 , where q is the connection integer ,this number gives the taps and is defined by:

$q=2q_1+2^2q_2+2^4q_4+\ldots\ldots\ldots+2^nq_n-1$,q-must prime .

1.10.encryption with FCSR :

the binary message stream $p=p_1p_2\ldots\ldots\ldots$, is encipher by computing $c_i=p_i\oplus k_i$,as the bits of the key stream(see figure(9)),



Figure(9)

1.11.Example:-

If we take a 3-bit FCRS at the first and second bit ,its initial value is 001 ,and the initial contents of the carry register is 0,then

K_i=10010111010………

To encipher the message

p_i=11011101011….

$c_i=p_i\oplus k_i$

=01001010001…….

And decipher is

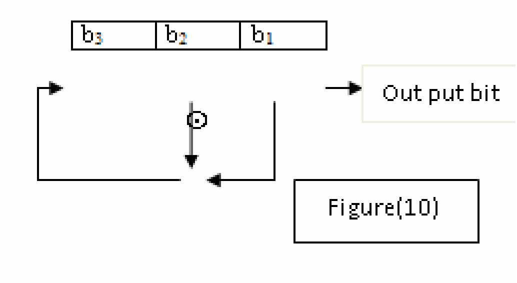
$p_i=c_i\oplus k_i$

=11011101011…….

1.12.Nonlinear-feedback shift registers:

It is easy to imagine a more complicated feedback sequence than the ones used in $LFSR_s$ or $FCSR_s$,the problem is that there isn't any mathematical theory that can analyze them ,in particular there is some problems with nonlinear-feedback shift register sequence[cf.[5,6,7]]

As ,a 3-bit shift register with the following feedback function the new bit is the first bit times the second bit ,if it is initialized with the value 110 ,it produces the following sequence of internal states,



Figure(10)

1 1 0

0 1 1

1 0 1

0 1 0

0 0 1

0 0 0

0 0 0

.

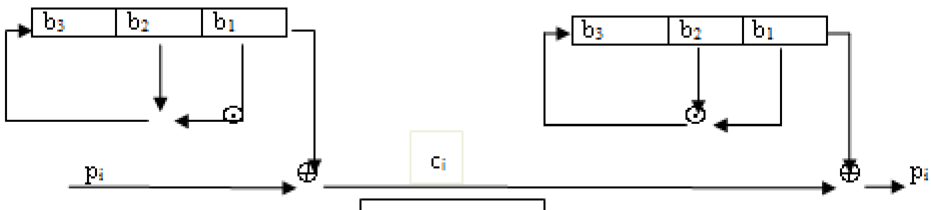
.

And so forever ,the output sequence is the string of least significant bits :

K_i=01101000000…………

1.13.Encryption with nonlinear FSR:

The binary message stream $p=p_1p_2$………. ,is encipher by computing $c_i=p_i \oplus k_i$,as the bits of key stream ,



Figure(11)

1.14.Example:

If we take a 3-bit nonlinear FSR at the first bit times the second bit ,if it is initialized with the value 110 ,then

K_i=0110100000…………

To encipher the message

P_i=1101011101……..

$c_i=p_i \oplus k_i$

=1011111101…..

And decipher is

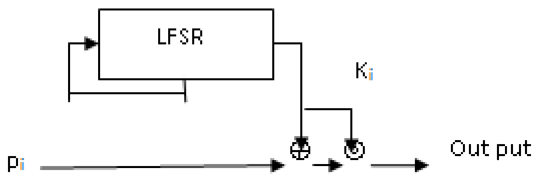
$p_i=c_i \oplus k_i$

=1101011101………..

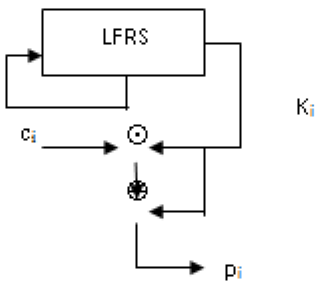
2.On adaptive of generations:

2.1.$\mathcal{A}$ - Linear feedback shift registers:

The binary message stream $p=p_1p_2$……, is enciphered by computing $c_i=p_i \oplus k_i$,as the bits of the key stream are generated and by using $\mathcal{E}$-$\mathcal{A}$ and $\mathcal{D}$-$\mathcal{A}$ laws we can develop the generating for more complexity, where the encipher by using the $\mathcal{E}$-$\mathcal{A}$ law and the deciphering is done by using $\mathcal{D}$-$\mathcal{A}$ law (see figure (12)) .
Where
$\mathcal{E}$-$\mathcal{A}$=min(max(1-c,k),max(c,1-k))
$\mathcal{D}$-$\mathcal{A}$=min(max(1- $\mathcal{E}$-$\mathcal{A}$,k),max($\mathcal{E}$-$\mathcal{A}$,1-k))
The encryption



And decryption



Figure(12)

2.2.Example:

If we take a 4-stage LFSR with tap sequence T=(1001) and initial state 1010,we have

K=010110010001111….

To encipher message p_i=00101010.……

$c_i=p_i \oplus k_i$ =01110011………

$\mathcal{E}$-$\mathcal{A}$=min(max(1-c,k),max(c,1-k))
=min(max(1-0,0),max(0,1-0)

=min(1,1)

=1

And another states we can computes ,and so that

The sender is ,

$\mathcal{E}$-$\mathcal{A}$=11010101……….

And ,to decipher the cipher text

$\mathcal{D}$-$\mathcal{A}$=min(max(1- $\mathcal{E}$-$\mathcal{A}$,k),max($\mathcal{E}$-$\mathcal{A}$,1-k))
=min(max(1-1,0),max(1,1-0))

=min(0,1)

=0

And another states we can computes ,and so that

$\mathcal{D}$-$\mathcal{A}$=01110011…………..

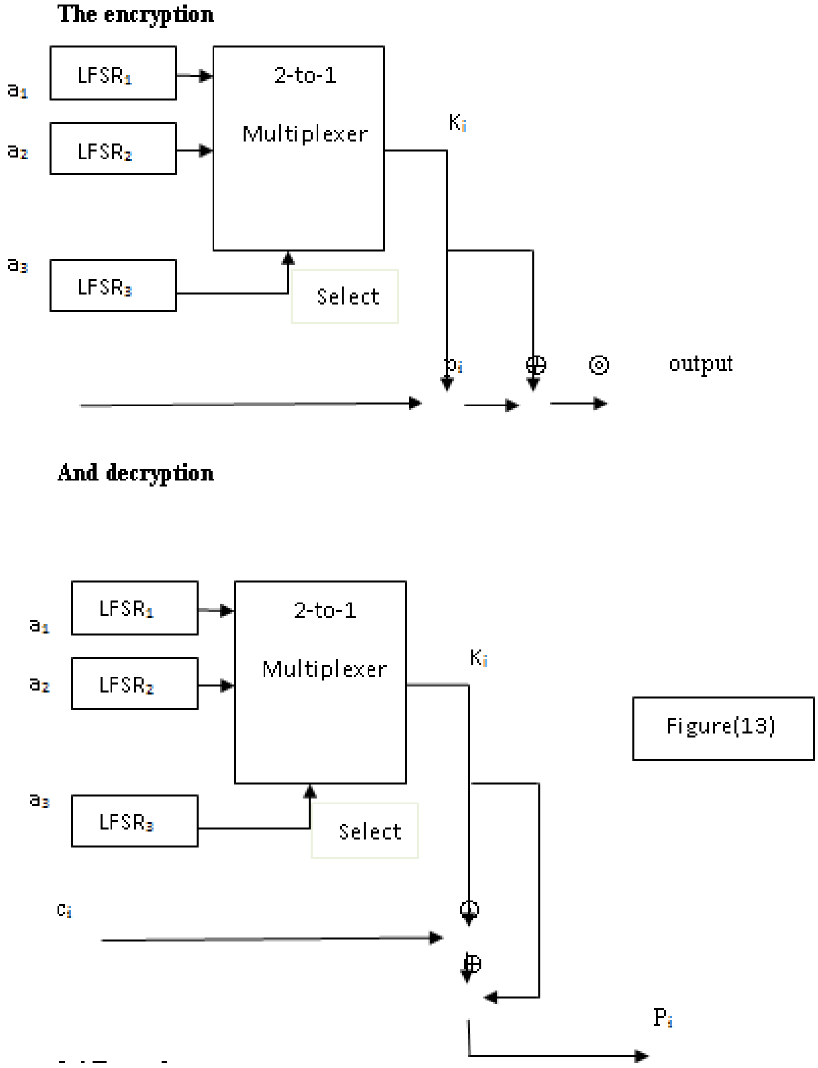
p_i= **$\mathcal{D}$-$\mathcal{A}$** $\oplus k_i$

=00101010……………

2.3.$\mathcal{A}$ -Geffe Generator:

The binary message stream $p=p_1p_2$……, is enciphered by computing $c_i=p_i \oplus k_i$,as the bits of the key stream are generated and by using **$\mathcal{E}$-$\mathcal{A}$** and **$\mathcal{D}$-$\mathcal{A}$** laws we can develop the generating for more complexity, where the encipher by using the **$\mathcal{E}$-$\mathcal{A}$** law and the deciphering is done by using **$\mathcal{D}$-$\mathcal{A}$** law (see figure(13)).

The encryption



2.4.Example:

If we take a 4-stage $LFSR_1$,$LFSR_2$ and $LFSR_3$ with the same tap sequence T=(1001),with initial state (1101),(0011) and(0101) respectively ,then the key stream is

a_1=10110010…………

a_2=11001000…………

a_3=10101100…………

$-a_3$=01010011………..

and

K_i=11100000…………

Now to encipher the message ,p_i=10010101………

$c_i=p_i \oplus k_i$

=01110101………

$\mathcal{E}$-$\mathcal{A}$=min(max(1-c,k),max(c,1-k))

=min(max(1-0,1),max(0,1-1)

=min(1,0)

=0

And another states we can computes ,and so that

The sender is , **$\mathcal{E}$-$\mathcal{A}$=01101010……………**

And ,to decipher the cipher text

$\mathcal{D}$-$\mathcal{A}$=min(max(1- $\mathcal{E}$-$\mathcal{A}$,k),max($\mathcal{E}$-$\mathcal{A}$,1-k))

=min(max(1-0,1),max(0,1-1))

=min(1,0)

=0

And another states we can computes ,and so that

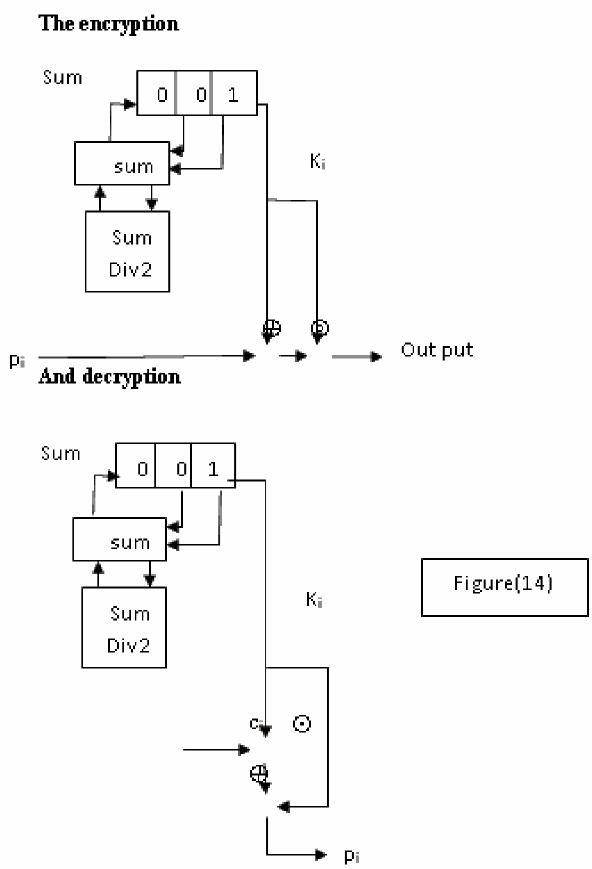
$\mathcal{D}$-$\mathcal{A}$=01110101………..

p_i= **$\mathcal{D}$-$\mathcal{A}$** $\oplus k_i$

=10010101…….

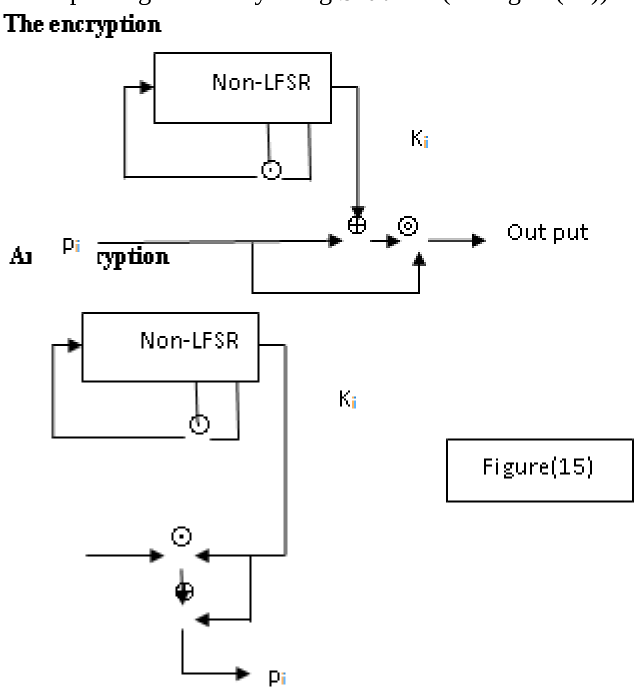
2.5.$\mathcal{A}$ -Feedback with carry shift registers:

Now to develop the FCSR we use the **$\mathcal{E}$-$\mathcal{A}$** and **$\mathcal{D}$-$\mathcal{A}$** laws, where the encipher by using the **$\mathcal{E}$-$\mathcal{A}$** law and the deciphering is done by using **$\mathcal{D}$-$\mathcal{A}$** law (see figure(14)) .



2.6.Example:
If we take a 3-bit FCRS at the first and second bit ,its initial value is 001 ,and the initial contents of the carry register is 0,then
K_i=10010111010………
To encipher the message
P_i=1010010101…..
$C_i=p_i \oplus k_i$ =0011001000……
$\boldsymbol{\mathcal{E}\text{-}\mathcal{A}=\min(\max(1-c,k),\max(c,1-k))}$
$=\min(\max(1-0,1),\max(0,1-1))$
$=\min(1,0)$
$=0$
And another states we can computes ,and so that
The sender is ,
$\mathcal{E}\text{-}\mathcal{A}$=0101101010………
And ,to decipher the cipher text
$\boldsymbol{\mathcal{D}\text{-}\mathcal{A}=\min(\max(1-\mathcal{E}\text{-}\mathcal{A},k),\max(\mathcal{E}\text{-}\mathcal{A},1-k))}$
$=\min(\max(1-0,1),\max(0,1-1))$
$=\min(1,0)$
$=0$
And another states we can computes ,and so that
$\mathcal{D}\text{-}\mathcal{A}$=0011001000………..
$p_i = \mathcal{D}\text{-}\mathcal{A} \oplus k_i$
=1010010101…….

2.7.$\mathcal{A}$ -Nonlinear-feedback shift registers:
Now to develop the Nonlinear-feedback shift registers we use the $\mathcal{E}\text{-}\mathcal{A}$ and $\mathcal{D}\text{-}\mathcal{A}$ laws, where the encipher by using the $\mathcal{E}\text{-}\mathcal{A}$ law and the deciphering is done by using $\mathcal{D}\text{-}\mathcal{A}$ law (see figure(15)) .



2.8.Example:
If we take a 3-bit nonlinear FSR at the first bit times the second bit ,if it is initialized with the value 110 ,then
K_i=0110100000…………
To encipher the message
P_i=10110101……..
$c_i=p_i \oplus k_i$=11011101…..
$\boldsymbol{\mathcal{E}\text{-}\mathcal{A}=\min(\max(1-c,k),\max(c,1-k))}$
$=\min(\max(1-1,0),\max(1,1-0))$
$=\min(0,1)$
$=0$
And another states we can computes ,and so that

The sender is ,
$\mathcal{E}$-$\mathcal{A}$=01001010..………
And ,to decipher the cipher text

$\mathcal{D}$-$\mathcal{A}$=min(max(1- $\mathcal{E}$-$\mathcal{A}$,k),max($\mathcal{E}$-$\mathcal{A}$,1-k))
=min(max(1-0,0),max(0,1-0))
=min(1,1)
=1

And another states we can computes ,and so that
$\mathcal{D}$-$\mathcal{A}$=11011101………..
p_i= $\mathcal{D}$-$\mathcal{A}$ $\oplus k_i$=10110101…….

References:

[1]- A. Menezes, P. van Oorschot, and S. Vanstone" Handbook of Applied Cryptography", CRC Press, Inc.; 1997.

[2]-Bruce,"application cryptography",second edition,published by john wiley and sons,inc,p.369-413,1996.

[3]-Jennifer S. and Josef p."cryptography: an introduction to computer security", by prentice hall of astralia pty,lid , 1982.

[4] -H. van Tilborg,"Coding Theory at Work in Cryptology and Vice Versa. Handbook of Coding Theory", vol. 2. Ed. by V.S. Pless and W.C. Hu_man. Elsevier Science B.V., 1998.

[5]- Fran¸cois Arnault, Thierry P. Berger, and Marine Minier," Some Results on FCSR Automata With Applications to the Security of FCSR-Based Pseudorandom Generators. IEEE Transactions on Information Theory", 2008.

[6]- Mark Goresky," Fibonacci and Galois Representations of Feedback-With-Carry Shift Registers",IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 48, NO. 11, NOVEMBER 2002

[7]- Maria George and Peter Alfke"Linear Feedback Shift Registersin Virtex Devices",XAPP210 (v1.3) April 30, 2007.