6. Stream Cipher Cryptosystem

6.1.Stream Ciphers

There are two types of stream ciphers - one for which the key-bit stream is independent of the plaintext and the other for which the key-bit stream is a function of either the plaintext or the ciphertext depending on the feedback connection. The former is often called a ***synchronous cipher*** because it requires synchronization between the key-bit stream and the ciphertext for successful decipherment. The later is often referred to as a ***self-synchronizing cipher*** or auto-key cipher because an erroneously added or deleted bit causes only a fixed number of errors in the deciphered plaintext, after which the correct plaintext is again restored.

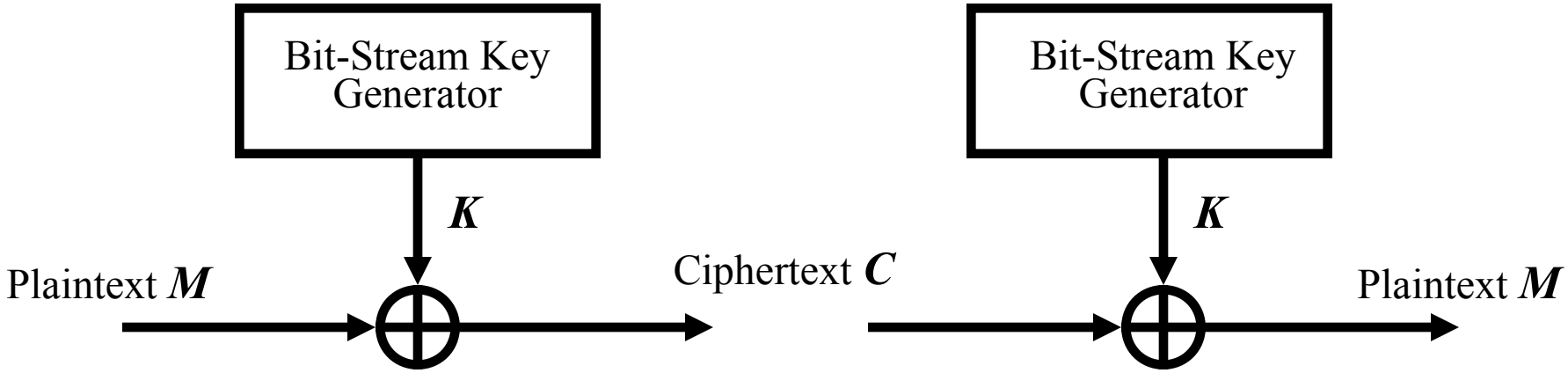


Fig. 6.1. Stream Cipher

6. Stream Cipher Cryptosystem

6.1. Stream Cipher

Some of the differences between block and stream cipher are:

1.The block cipher enciphers a single data block at one time, but the stream cipher requires to encipher or decipher on a bit-by-bit bases;

2.In the block cipher, every ciphertext bit is a complex function of plaintext and key bits via intersymbol dependence, but in the stream cipher every cipher bit is produced by Exclusive Or operation of the plaintext bit and key bit;

3.The block cipher may not require an initial seed vector, but the stream cipher does;

4.Block cipher like DES used in the commercial sector, but stream ciphers are used for military purposes.

The main part of any stream cipher is key generation algorithm. More often used algorithm for various application is ***Multiplicative Congruent Algorithm***

$$x_{t+1}=(ax_t+c) \bmod N;$$

Where for Pentium PC is a very popular module $N=2^{31}-1=2147483647,$ and "magic" values of *a* are: *16807,630360016, 1078318381, 1203248318, 397204094, 2037812808, 1323257245, 764261123, 112817, ...*

6. Stream Cipher Cryptosystem

6.1. Stream Cipher

Pseudorandom Number Generators

1. $x_{t+1}=(1176x_t+1476x_{t-1}+1776x_{t-2}) \bmod (2^{32}-5)$;

2. $x_{t+1}=(2^{13}(x_t+x_{t-1}+x_{t-2})) \bmod (2^{32}-5)$;

3. $x_{t+1}=(1995x_t+1998x_{t-1}+2001x_{t-2}) \bmod (2^{32}-849)$;

4. $x_{t+1}=(2^{19}(x_t+x_{t-1}+x_{t-2})) \bmod (2^{32}-1629)$;

5. $x_{t+1}=(5115x_t+1776x_{t-1}+1492x_{t-2}+2111111111x_{t-3}+c_t) \bmod 2^{32}$;
$c_t=\lfloor (5115x_{t-1}+1776x_{t-2}+1492x_{t-3}+2111111111x_{t-4}+c_{t-1})/ 2^{32} \rfloor$.

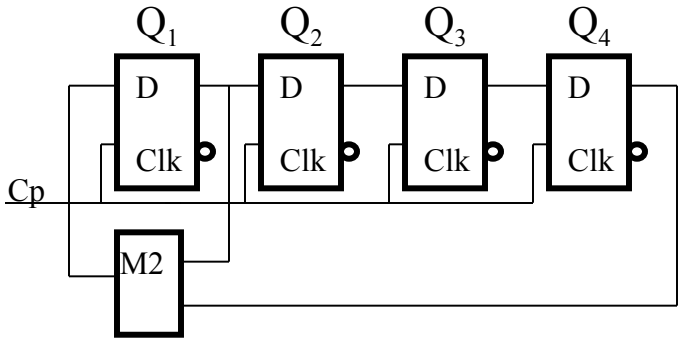
6. COMBO: $z_n=(x_n+y_n) \bmod 2^{32}$, $x_n=(x_{n-1}*x_{n-2}) \bmod 2^{32}$, $y_n=(y_{n-1}+c_n) \bmod 2^{16}$, $c_n=\lfloor (y_{n-1}+c_{n-1}) /2^{16} \rfloor$.

7. **KISS:** $z_n=(x_n+y_n+u_n) \bmod 2^{32}$, $x_n=(69069x_{n-1}+1) \bmod 2^{32}$, $y_n=y_{n-1}(I_{32}+L^{13})(I_{32}+R^{17})(I_{32}+L^5)$, $u_n=(2u_{n-1}+u_{n-2}+c_n) \bmod 2^{32}$, $c_n=\lfloor (2u_{n-2}+u_{n-3}+c_{n-1}) / \bmod 2^{32} \rfloor$.

6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

A ***linear feedback shift register (LFSR)*** is often used as the cryptographic key encoder for a stream cipher system. One of the main reason for this is that LFSR is easily obtainable and comparatively inexpensive. The encoder that generates the key-bit stream must be deterministic so that the key-bit stream may be reproduced for decipherment According to the primitive polynomial $\varphi(x)=1+x+x^4$ the following block diagram of the LFSR can be design



$$Q_1(k+1) = Q_1(k) \oplus Q_4(k)$$

$$Q_2(k+1) = Q_1(k)$$

$$Q_3(k+1) = Q_2(k)$$

$$Q_4(k+1) = Q_3(k)$$

Fig. 6.2. LFSR for the generating polynomial $\phi(x)=1+x+x^4$

State sequence for LFSR

State	$Q_1Q_2Q_3Q_4$	State	$Q_1Q_2Q_3Q_4$
0	1 0 0 0	8	1 1 0 1
1	1 1 0 0	9	0 1 1 0
2	1 1 1 0	10	0 0 1 1
3	1 1 1 1	11	1 0 0 1
4	0 1 1 1	12	0 1 0 0
5	1 0 1 1	13	0 0 1 0
6	0 1 0 1	14	0 0 0 1
7	1 0 1 0	15	1 0 0 0

6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

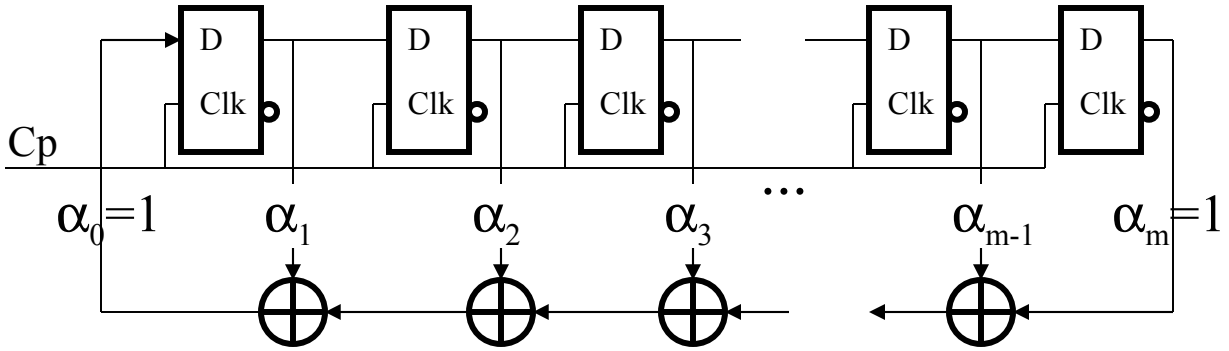
One primitive polynomial for each *m* from *1* to *28*

$m=\deg\varphi(x)$	$\varphi(x)$	$m=\deg\varphi(x)$	$\varphi(x)$
1	$1+x$	15	$1+x+x^{15}$
2	$1+x+x^2$	16	$1+x^2+x^3+x^5+x^{16}$
3	$1+x+x^3$	17	$1+x^3+x^{17}$
4	$1+x+x^4$	18	$1+x^7+x^{18}$
5	$1+x^2+x^5$	19	$1+x+x^2+x^5+x^{19}$
6	$1+x+x^6$	20	$1+x^3+x^{20}$
7	$1+x+x^7$	21	$1+x^2+x^{21}$
8	$1+x+x^5+x^6+x^8$	22	$1+x+x^{22}$
9	$1+x^4+x^9$	23	$1+x^5+x^{23}$
10	$1+x^3+x^{10}$	24	$1+x^3+x^4+x^{24}$
11	$1+x^2+x^{11}$	25	$1+x^3+x^{25}$
12	$1+x^3+x^4+x^7+x^{12}$	26	$1+x+x^2+x^6+x^{26}$
13	$1+x+x^3+x^4+x^{13}$	27	$1+x+x^2+x^5+x^{27}$
14	$1+x+x^{11}+x^{12}+x^{14}$	28	$1+x^3+x^{28}$

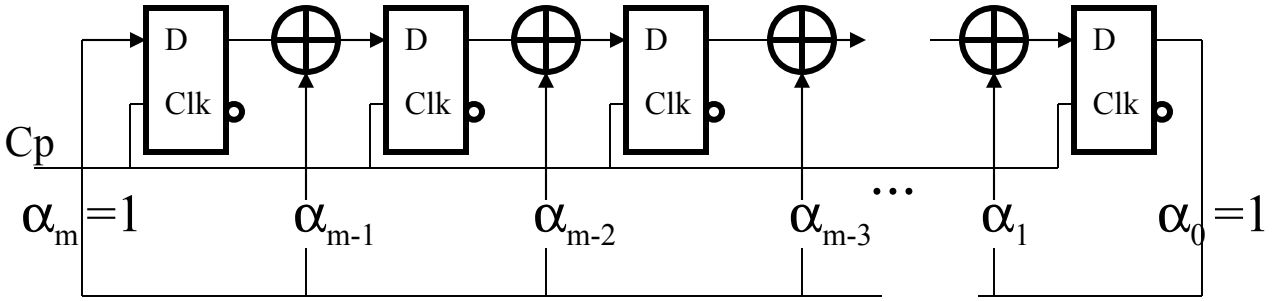
6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

$$\varphi(x)=\alpha_0+\alpha_1 x+\alpha_2 x^2+\ldots+\alpha_{m-1}x^{m-1}+\alpha_m x^m;\ \alpha_m=\alpha_0=1;\ \alpha_i\in\{0,1\}$$



Standard form of LFSR Type 1

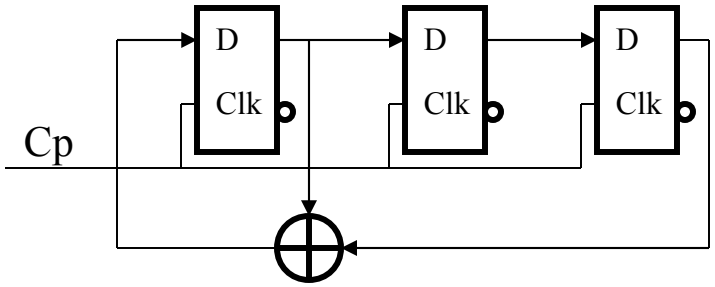


Standard form of LFSR Type 2

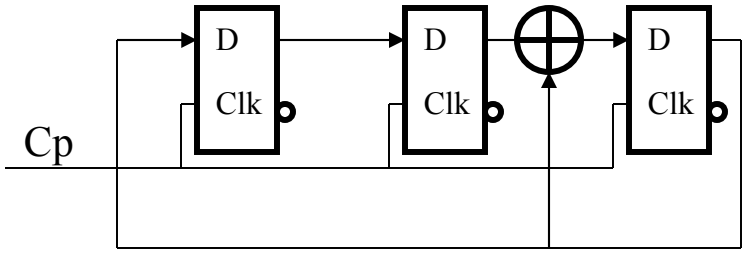
6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

Example $\varphi(x)=1+x+x^3$



1 0 0
1 1 0
1 1 1
0 1 1
1 0 1
0 1 0
0 0 1



1 0 0
0 1 0
0 0 1
1 0 1
1 1 1
1 1 0
0 1 1

6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

Mathematical description of LFSR

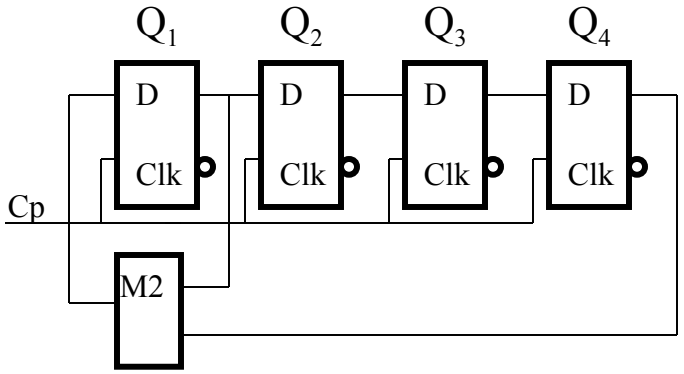
$$\begin{vmatrix} a_1(k+1) \\ a_2(k+1) \\ a_3(k+1) \\ \dots \\ a_m(k+1) \end{vmatrix} = \begin{vmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{m-1} & \alpha_m \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{vmatrix} \times \begin{vmatrix} a_1(k) \\ a_2(k) \\ a_3(k) \\ \dots \\ a_m(k) \end{vmatrix}$$

$$A(k+1) = V \times A(k)$$

$$a_1(k+1) = \sum_{i=1}^{m} \alpha_i a_i(k);$$

$$a_j(k+1) = a_{j-1}(k), j = \overline{2,m}, k = 0,1,2,\dots$$

Example



$$V = \begin{vmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix}$$

6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

Properties of pseudorandom sequences (M-sequences)

1. There are $\Phi(2^m-1)/m$ primitive polynomial for a given m, where Φ is a Euler function.
2. For a given polynomial $\varphi(x)$ there is an inverse polynomial which can be derived according to the following relation $\varphi^{-1}(x)=x^m\varphi(x^{-1})$.
3. The period L of a M-sequence depends on of the primitive polynomial degree only $L=2^m-1$.
4. For the specified polynomial $\varphi(x)$ there are L distinct M-sequences that are differ by phase shifts. Thus 15 M-sequences are associated with the polynomial $\varphi(x)=1+x+x^4$.
5. In M-sequence there is unpredictable (pseudorandom) sequence of ones and zeros. Their probabilities is given by

$$p(a_k=1)=\frac{2^{m-1}}{2^m-1}=\frac{1}{2}+\frac{1}{2^{m+1}-2};$$

$$p(a_k=0)=\frac{2^{m-1}-1}{2^m-1}=\frac{1}{2}-\frac{1}{2^{m+1}-2};$$

6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

6. There is one run of ***m*** (consecutive) 1's and one run of ***m-1*** 0's. For ***m-1>r>0***, there are $2^{m-(r+2)}$ runs of length ***r*** for 1's and the same number of runs of 0's. For the M-sequence ***000111101011001*** generated according to the polynomial $\varphi(x)=1+x+x^4$ there are one run of 4 1's one run of 3 0's, one run each of 2 0's or 1's.

7. There is the autocorrelation property that measures the similarity between original M-sequence and a shifted version of the same sequence. Any pair of such sequence will be identical in $2^{m-1}-1$ positions and will differ in 2^m-1 positions.

000111101011001
011110101100100

8. ***'Shift-and-add'*** property. For any ***s (1≤s<L)*** there are exist an ***r ≠s (1≤r<L)*** such that $\{a_k\}\oplus\{a_{k-s}\}=\{a_{k-r}\}$.

$\{a_0\}$	***000111101011001***
$\{a_{-2}\}$	***011110101100100***
$\{a_{-9}\}$	***011001000111101***

6. Stream Cipher Cryptosystem

6.2. LFSR-Linear Feedback Shift Register

9. Among the ***L*** M-sequences generated by the polynomial $\varphi(x)$ there exists a single sequence with the property $a_k=a_{2k}$, $k=0,1,2,\ldots$ which is called ***characteristic*** and derived as follows. For the specified polynomial the system of linear equations

$$a_i=a_{2i};\ i=0 \text{ to } m-1;$$

As an example, we shall fined characteristic sequence for the generating polynomial $\varphi(x)=1+x^3+x^4$. For this case the system of linear equations takes the form

$$a_0=a_0$$

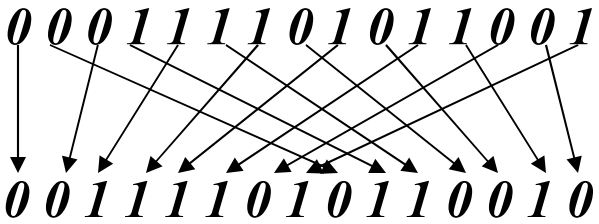
$$a_1=a_2$$

$$a_2=a_4=a_0 \oplus a_1$$

$$a_3=a_6=a_2 \oplus a_3$$

The solution is $a_1a_2a_3a_4=1000$.

10. Property of ***'Decimation'***. Decimation of the M-sequence $\{a_i\}$ by q, $(q=1,2,3,\ldots)$ means the generation of another sequence $\{b_j\}$ as qth elements of original M-sequence $\{a_i\}$, i.e. $b_j=a_{qj}$. With (L,q)=1 the period of $\{b_j\}$ is 2^m-1, and decimation is said to be proper or normal.



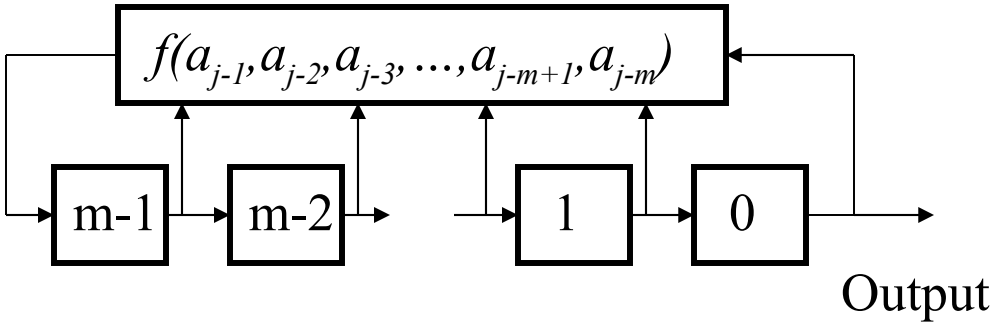
xxxx:

6. Stream Cipher Cryptosystem

6.3. *NFSR - NonLinear Feedback Shift Registers*

A general feedback shift register (FSR) of length m consists of m stages (or delay elements). During each unit of time the following operations are performed: (i) the contents of stage 0 is output and forms part of the output sequence; (ii) the contents of stage j is moved to stage $j-1$ for each j, $1 \le j \le m-1$; (iii) the new content of stage $m-1$ is the feedback bit $a_j = f(a_{j-1}, a_{j-2}, a_{j-3}, \ldots, a_{j-m+1}, a_{j-m})$, where the feedback function f is a Boolean function and a_{j-r} is the previous content of the shift register.

Not that if the feedback function f is a linear function, then the FSR is an LFSR, otherwise the FSR is called a nonlinear FSR.



Example 6.1. De Bruin sequence for the case of $m=3$ and nonlinear feedback function $f(x_1, x_2, x_3) = 1 \oplus x_2 \oplus x_3 \oplus x_1 x_2$.

The output sequence is the ***de Bruijn*** sequence with cycle *0,0,0,1,1,1,0,1*

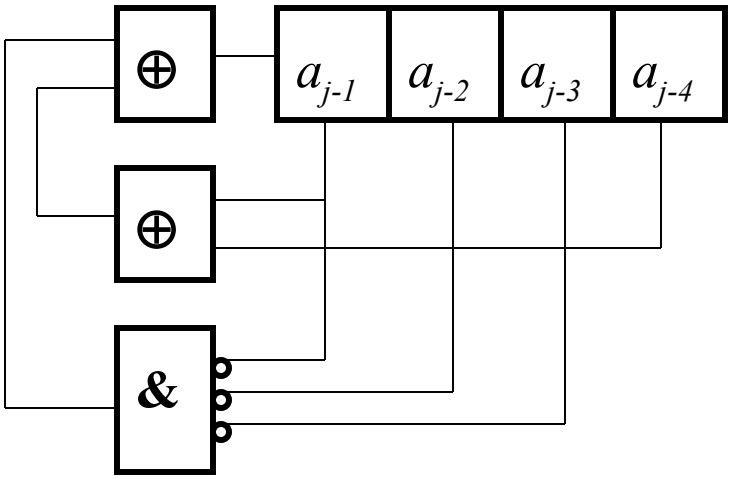
6. Stream Cipher Cryptosystem

6.3. NFSR - NonLinear Feedback Shift Registers

Converting a maximum-length LFSR to a de Bruin FSR

Let we have a LFSR of length m with (linear) feedback function $a_j=f(a_{j-1},a_{j-2},a_{j-3},\ldots,a_{j-m+1},a_{j-m})$, then FSR with feedback function $g(a_{j-1},a_{j-2},a_{j-3},\ldots,a_{j-m+1},a_{j-m})=f(a_{j-1},a_{j-2},a_{j-3},\ldots,a_{j-m+1},a_{j-m})\oplus a^*_{j-1}a^*_{j-2}a^*_{j-3}\ldots a^*_{j-m+1}$ is a ***de Bruijn FSR.*** Here a^*_{j-1} denotes the complement of a_{j-1}.

Example 6.2. De Bruin sequence for the case of $m=3$ and linear feedback function described by primitive polynomial $\varphi(x)=1\oplus x^1\oplus x^4$, has a nonlinear function $g(a_1,a_2,a_3,a_4)=a_1\oplus a_4\oplus a^*_1a^*_2a^*_3$.



a_3	a_2	a_1	a_0
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
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
0	***0***	***0***	***0***
1	0	0	0

6. Stream Cipher Cryptosystem

6.3. NFSR - NonLinear Feedback Shift Registers

Ford sequence

Ford sequences are a special class of de Bruijn sequences and due to the simplest algorithm for generation can be applied for bit stream generation.

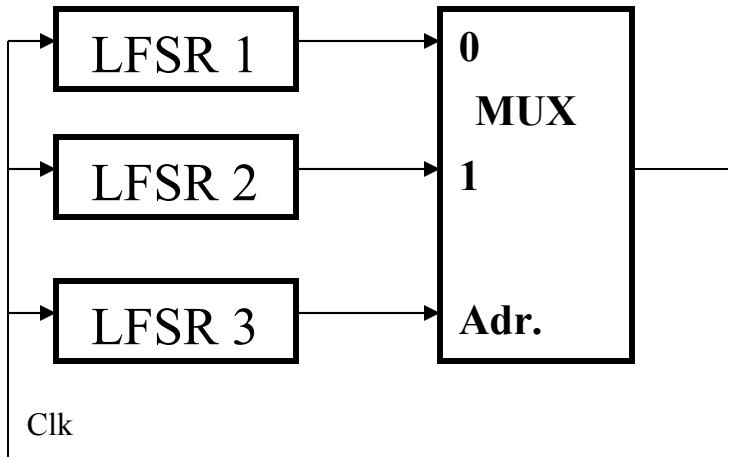
Algorithm: A successive m-bit code $X_k=(b_2,b_3,b_4,\ldots,b_{m+1})$ is generated on the basis of the preceding code $X_{k-1}=(b_1,b_2,b_3,\ldots,b_m)$, where the value of $b_{m+1}\in\{0,1\}$ is determined as follows. The code of the form $X'_{k-1}=(b_2,b_3,b_4,\ldots,b_m,1)$ is generated and all its cyclic shifts are obtained, among which code $X''_{k-1}=(b_i,b_{i+1},b_{i+2},\ldots,b_m,1,b_2,b_3,\ldots,b_{i-1})$, which is the largest m-bit number is chosen. If $b_2=b_3=\ldots=b_{i-1}=0$, *then* $b_{m+1}=b_1\oplus 1$, otherwise $b_{m+1}=b_1$.

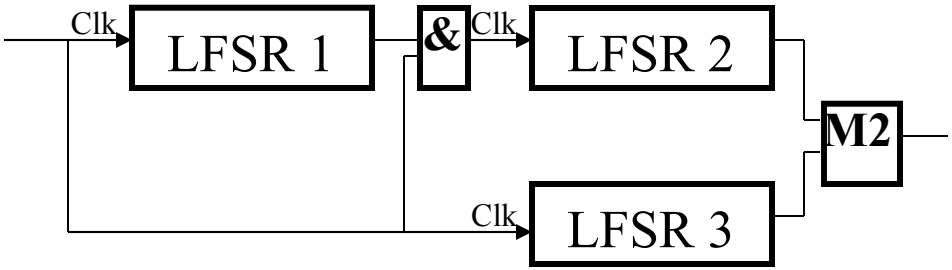
Example 6.3. Ford sequence for the case of $m=4$

k	X_k	X''_{k-1}	b_{m+1}
1	0000	1000	
2	0001	1100	$b_1\oplus 1=0\oplus 1=1$
3	0011	1110	$b_1\oplus 1=0\oplus 1=1$
4	0111	1111	$b_1\oplus 1=0\oplus 1=1$
5	1111	1111	$b_1\oplus 1=0\oplus 1=1$
6	1110	1110	$b_1\oplus 1=1\oplus 1=0$
7	1101	1110	$b_1=1$
8	1011	1110	$b_1=1$
9	0110	1110	$b_1\oplus 1=1\oplus 1=0$
10	1100	1100	$b_1=0$
11	1001	1100	$b_1=1$
12	0010	1010	$b_1\oplus 1=1\oplus 1=0$
13	0101	1110	$b_1\oplus 1=0\oplus 1=1$
14	1010	1010	$b_1=0$
15	0100	1100	$b_1\oplus 1=1\oplus 1=0$

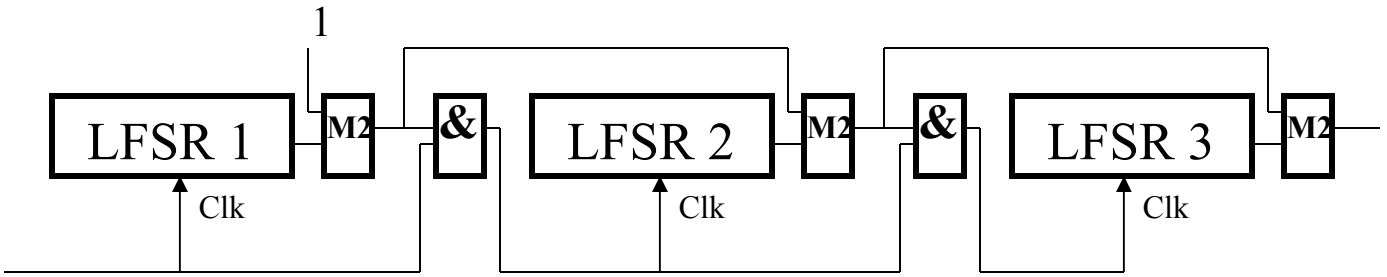
6. Stream Cipher Cryptosystem

6.3. NFSR - NonLinear Feedback Shift Registers





The ***Geffe generator***, as well as ***Stop and Go generator*** is defined by three maximum length LFSRs whose length *m1, m2* and *m3* are pair wise relatively prime, and the key stream length is $(2^{m1}-1)(2^{m2}-1)(2^{m3}-1)$.

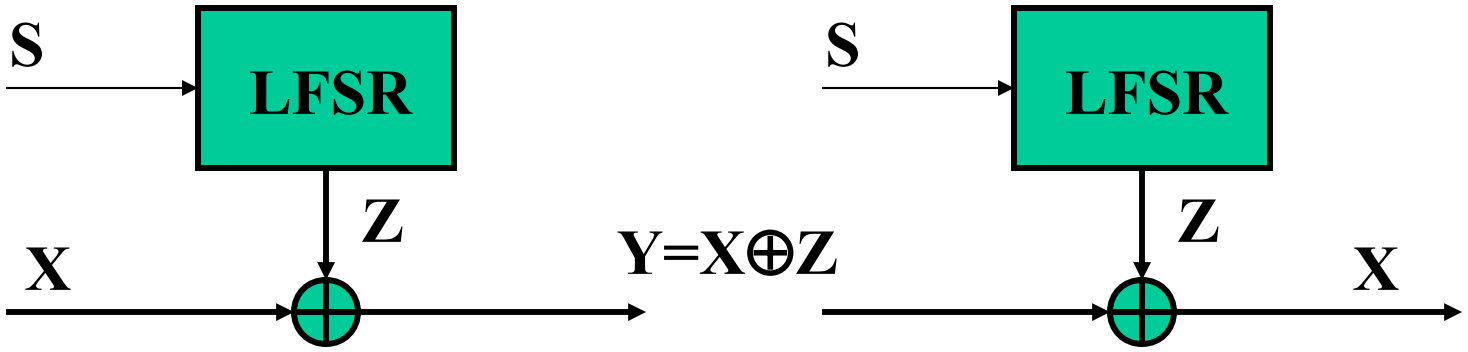


The ***Goldman Cascaded Key stream generator*** generates the sequence with the length is $(2^{m1}-1)(2^{m2}-1)(2^{m3}-1)$.

6. Stream Cipher Cryptosystem

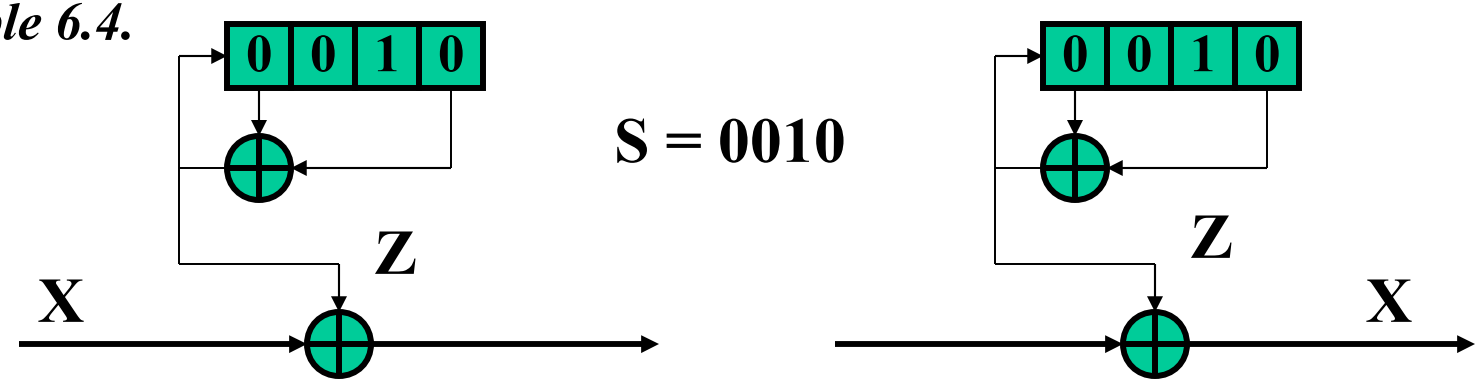
6.4. Synchronous Stream Cipher

Synchronous Stream Cipher can be implemented based on LFSR according to the following block diagram.



S – Seed Vector; **X** – Plaintext; **Y** – Ciphertext.

Example 6.4.



X=1010111111100001
Z=011110101100100
Y=110101010000101

6. Stream Cipher Cryptosystem

6.4. Synchronous Stream Cipher

However, the LFSR is not always suitable for key generation because an opponent can easily derive the entire key stream from ***2m*** bits of plaintext-ciphertext pairs.

$$b_{m+1} = \alpha_1 b_m \oplus \alpha_2 b_{m-1} \oplus \alpha_3 b_{m-2} \oplus \ldots \oplus \alpha_m b_1;$$

$$b_{m+2} = \alpha_1 b_{m+1} \oplus \alpha_2 b_m \oplus \alpha_3 b_{m-1} \oplus \ldots \oplus \alpha_m b_2;$$

$$b_{m+3} = \alpha_1 b_{m+2} \oplus \alpha_2 b_{m+1} \oplus \alpha_3 b_m \oplus \ldots \oplus \alpha_m b_3;$$

$$\ldots$$

$$b_{2m} = \alpha_1 b_{2m-1} \oplus \alpha_2 b_{2m-2} \oplus \alpha_3 b_{2m-3} \oplus \ldots \oplus \alpha_m b_m;$$

Example 6.4. Continuation. To brake stream cryptosystem shown on previous example first of all the pair of 8 bit plaintext-ciphertext have to be obtained. Let it be ***X=10101111*** and ***Y=11010101***. Then, the modulo two summation of this ***X*** an ***Y*** will result as a key stream ***Z***.

X	=	1	1	0	1	0	1	0	1
Y	=	1	0	1	0	1	1	1	1
Z	=	0	1	1	1	1	0	1	0
		b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8

$\Rightarrow$

$$b_5 = \alpha_1 b_4 \oplus \alpha_2 b_3 \oplus \alpha_3 b_2 \oplus \alpha_4 b_1;$$
$$b_6 = \alpha_1 b_5 \oplus \alpha_2 b_4 \oplus \alpha_3 b_3 \oplus \alpha_4 b_2;$$
$$b_7 = \alpha_1 b_6 \oplus \alpha_2 b_5 \oplus \alpha_3 b_4 \oplus \alpha_4 b_3;$$
$$b_8 = \alpha_1 b_7 \oplus \alpha_2 b_6 \oplus \alpha_3 b_5 \oplus \alpha_4 b_4;$$

$\Rightarrow$

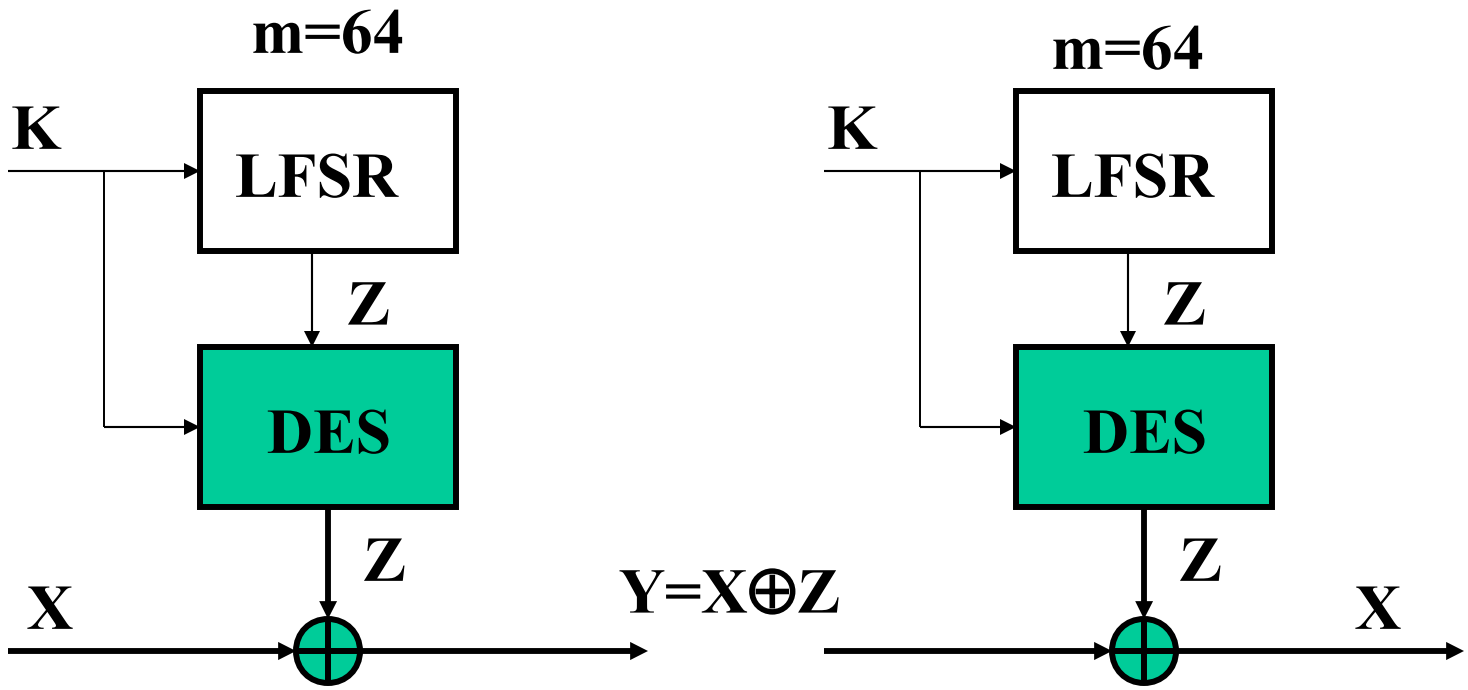
$$1 = \alpha_1 \oplus \alpha_2 \oplus \alpha_3;$$
$$0 = \alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4;$$
$$1 = \alpha_2 \oplus \alpha_3 \oplus \alpha_4;$$
$$0 = \alpha_1 \oplus \alpha_3 \oplus \alpha_4;$$

$$\alpha_4 = 1; \alpha_1 = 1; \alpha_2 = 0; \alpha_3 = 0;$$

6. Stream Cipher Cryptosystem

6.4. Synchronous Stream Cipher

To increase efficiency of the cryptosystem the combined approach can be used. As an example see below the combination of stream cipher and DES algorithm.

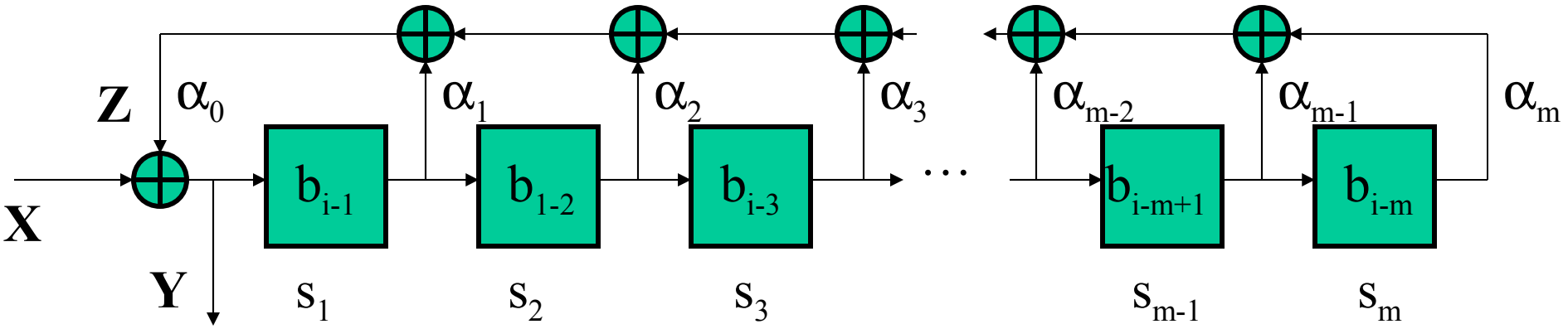


6. Stream Cipher Cryptosystem

6.4. Self-Synchronizing Stream Cipher

Self-synchronizing cipher systems are another class of stream cipher and are categorized as ***ciphertext autokey cipher*** and ***plaintext autokey cipher***.

Ciphertext Autokey Cipher



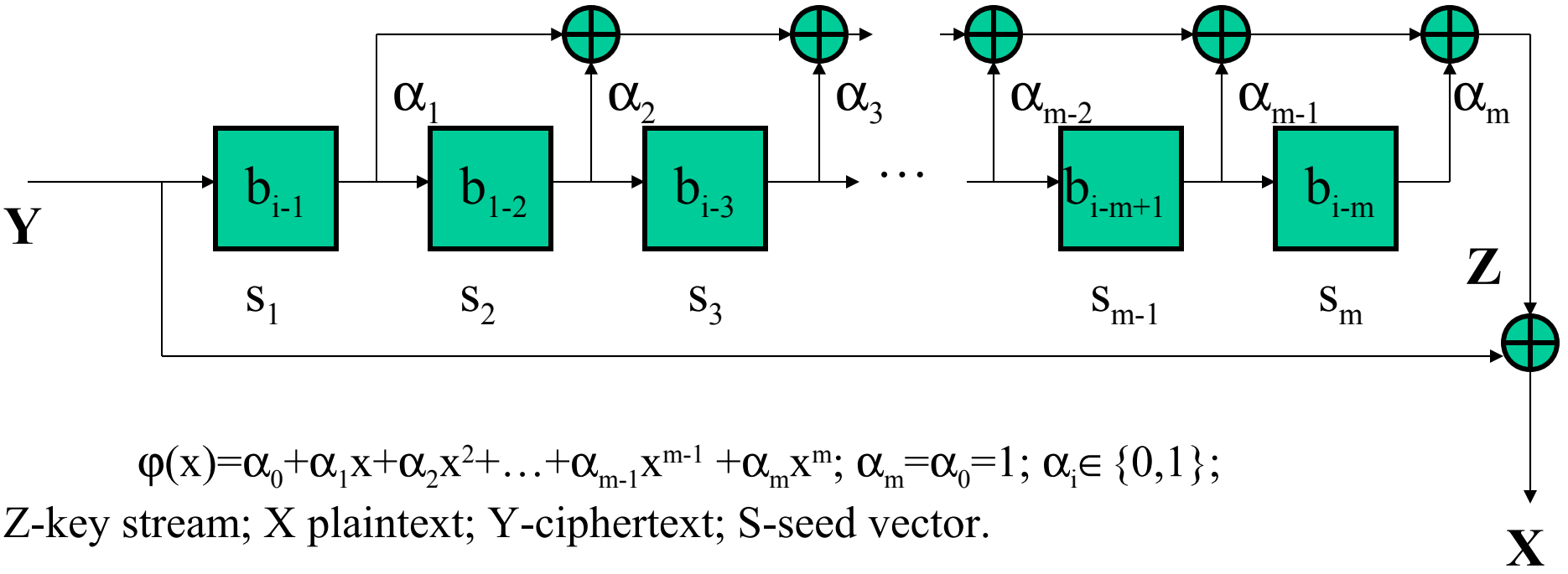
$\varphi(x)=\alpha_0+\alpha_1 x+\alpha_2 x^2+\ldots+\alpha_{m-1}x^{m-1}+\alpha_m x^m$; $\alpha_m=\alpha_0=1$; $\alpha_i \in \{0,1\}$; Z-key stream; X plaintext; Y-ciphertext; S-seed vector.

$$y_i = \begin{cases} x_i + \sum_{j=1}^{i} \alpha_j y_{i-j} + \sum_{j=i+1}^{m} \alpha_j s_{j-i}; & 0 \le i \le m-1; \\ x_i + \sum_{j=1}^{m} \alpha_j y_{i-j}; & i \ge m. \end{cases}$$

6. Stream Cipher Cryptosystem

6.4. Self-Synchronizing Stream Cipher

Ciphertext Autokey Decipher



$\varphi(x)=\alpha_0+\alpha_1 x+\alpha_2 x^2+\ldots+\alpha_{m-1}x^{m-1}+\alpha_m x^m$; $\alpha_m=\alpha_0=1$; $\alpha_i \in \{0,1\}$;

Z-key stream; X plaintext; Y-ciphertext; S-seed vector.

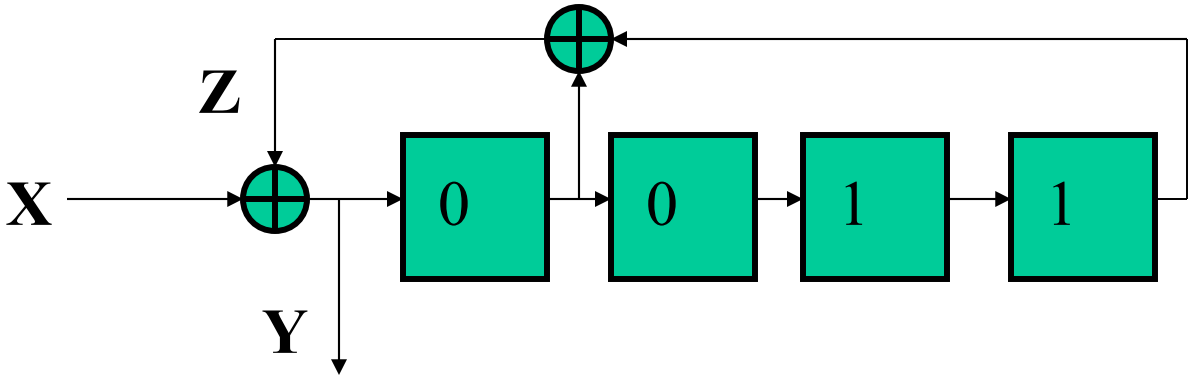
$$x_i = \begin{cases} y_i + \sum_{j=1}^{i} \alpha_j y_{i-j} + \sum_{j=i+1}^{m} \alpha_j s_{j-i}; & 0 \le i \le m-1; \\ y_i + \sum_{j=1}^{m} \alpha_j y_{i-j}; & i \ge m. \end{cases}$$

6. Stream Cipher Cryptosystem

6.4. Self-Synchronizing Stream Cipher

Example 6.5. $\varphi(x)=1+x+x^4$; $X=1110100$; $S=(s_1s_2s_3s_4)=0011$.

Ciphertext Autokey Cipher



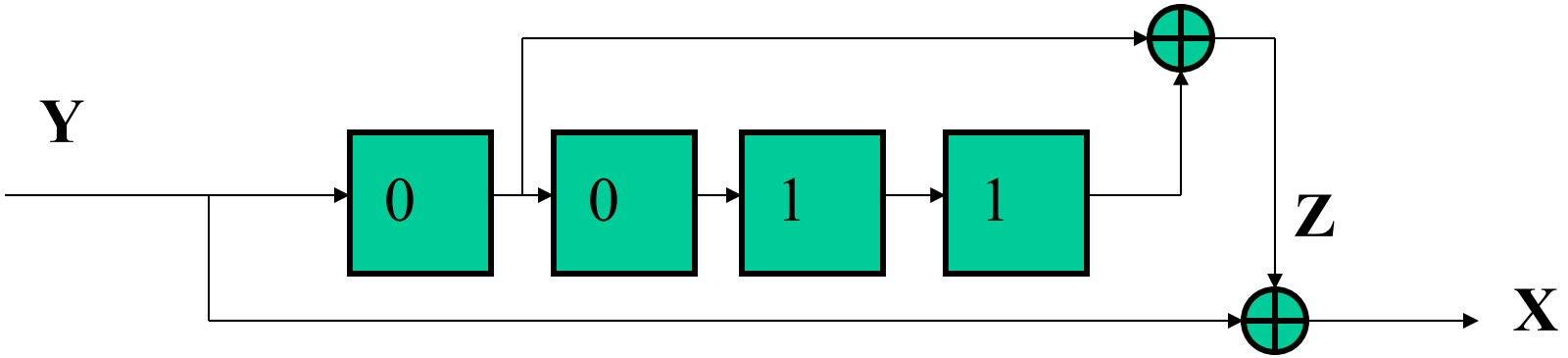
1	0	0	0	0	1
1	0	0	0	0	0
1	1	1	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
0	0	0	0	1	1
0	1	1	0	0	1
1	1	1	1	0	0

6. Stream Cipher Cryptosystem

6.4. Self-Synchronizing Stream Cipher

Example 6.5. Continuation. $\varphi(x)=1+x+x^4$; Y=00110011; $S=(s_1s_2s_3s_4)=0011$.

Ciphertext Autokey Decipher

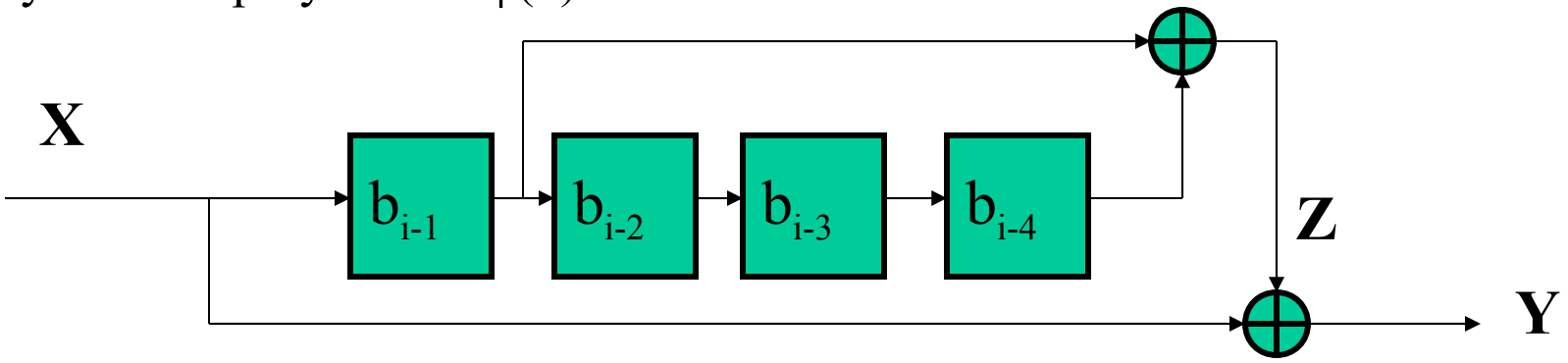


0	0	0	1	1	1
0	0	0	0	1	1
1	0	0	0	0	1
1	1	0	0	0	0
0	1	1	0	0	1
0	0	1	1	0	0
1	0	0	1	1	0
1	1	0	0	1	1

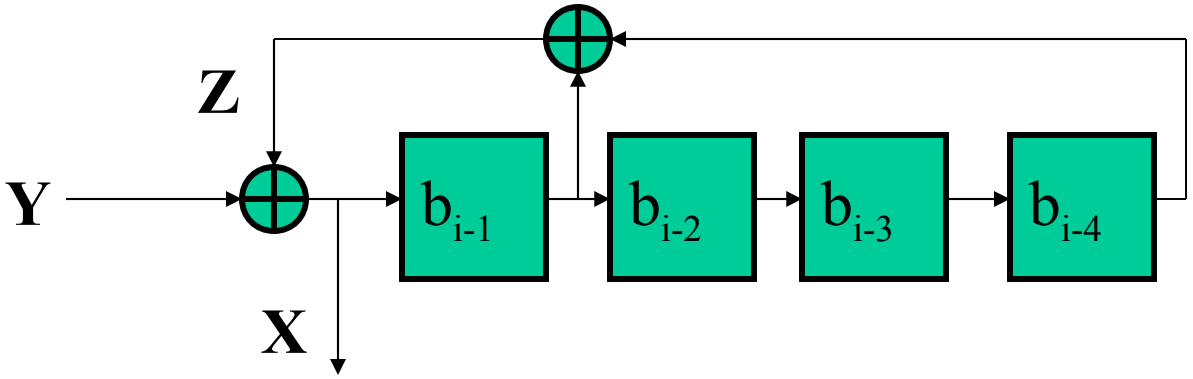
6. Stream Cipher Cryptosystem

6.4. Self-Synchronizing Stream Cipher

Plaintext autokey cipher are often called self-synchronizing cipher with plaintext input to the shift register. Let consider an example of such a type of the cryptosystem for polynomial $\varphi(x)=1+x+x^4$



A self-synchronizing encipher with plaintext excitation.



A autokey decipher system with plaintext feedback.

6. Stream Cipher Cryptosystem

6.4. SEAL-Software-optimized Encryption Algorithm

SEAL is a length-increasing pseudorandom function which maps 32-bit sequence number ***n*** to an ***L***-bit keystream under control of a 160-bit secret key ***a***. In the preprocessing stage, the key is stretched into tables using the table-generation function G_a; this function is based on the Secure Hash Algorithm SHA-1. There are SEAL 1.0 and SEAL 2.0, where SEAL 1.0 was base on the Secure Hash Algorithm (SHA) and SEAL 2.0 differ from SEAL 1.0 in the table-generation function for the former is based on the modified Secure Hash Algorithm SHA-1.***Plaintext autokey cipher*** are often called self-synchronizing cipher with plaintext input to the shift register. Let consider an example of such a type of the cryptosystem for polynomial $\varphi(x)=1+x+x^4$