Cryptography And Cryptanalysis

Ph. D. Course/ 2019-2020

Introduced By

Dr. Faez Hassan Ali

Lecture Three-2

Stream Cipher and Shift Register

Golomb and Massey Definitions for LFSR

Golomb Definition

Golomb considered that the SR is a r arrangement of memory, every memory contains the signal "on" (1) or "off" (0). Every cell shift its contain to the next cell in one time, and when there are no entry signals then the SR will be "off" after r shifting and to prevent that the SR must be re-fed in the 1st cell by sum xor some/all the contains of the SR. The values $a_{-1},a_{-2},\ldots,a_{-r}$ are the initial values of the SR. The output at time n, n=0,1,2,…, is:

$$a_n = C_1 a_{n-1} + C_2 a_{n-2} + \cdots C_r a_{n-r} = \sum_{i=1}^{r} C_i a_{n-i} \quad \ldots(1)$$

this relation called the Linear Recurrence Relation

and: $C(x) = 1 + \sum_{i=1}^{r} C_i x^i$ called characteristic polynomial of the LFSR.

Massey Definition

the code <C(D),L> denotes the LFSR with length L and has connection polynomial:

$$C(D)=1+C_1 D + C_2 D^2 + \cdots C_L D^L \quad \ldots(2)$$

Where D is the delay operator, if $c_i=1$ then the cell is take a part in connection function, else it is not. The initial value are $s_0,s_1,\ldots,s_{L-1}$ s.t. then the output (the output sequence) are $s_{L-1},\ldots,s_1,s_0$ when j<L but when j≥L then the output s_j can be obtained by the following recurrence relation (RR):

$$s_j = \sum_{i=1}^{L} C_i s_{j-i} \, , j \geq L \quad \ldots(3)$$

for RR (3) there is a Monic polynomial c(X) called characteristic polynomial of the LFSR.

$$c(X) = X^L + c_1 X^{L-1} + c_2 X^{L-2} + \cdots + c_L \in GF(2), GF(2)[X].$$

$c_i \in GF(2)$, GF(2)[X] is the ring of the all binary polynomials defined on GF(2).

Note that c(X) and C(D) are the reciprocal polynomial s.t.: $C(D) = D^L c(D^{-1})$ or $c(X)=X^L C(X^{-1})$.

Theorem (1): $c(x) = \sum_{i=0}^{L} c_i x^i$ is primitive (irreducible) polynomial iff $c'(x) = \sum_{i=0}^{L} c_i x^{L-i}$ is primitive (irreducible) polynomial.

Engineering Concepts of Shift Register

A **feedback shift register** (FSR) is made up of two parts: a SR and a **feedback function**. The SR is a sequence of bits, (the length of SR is figured in bits). Each time a bit is needed, all of the bits in the SR 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 SR is 1 bit, often the least significant bit. The period of a SR is the length of the output sequence before it starts repeating.

The simplest kind of feedback shift register is a **Linear Feedback Shift Register** (LFSR). The feedback function is simply the XOR of certain bits in the register; the list of these bits is called a **tap sequence**. Because of the simple feedback sequence, a large body of mathematical theory can be applied to analyzing LFSRs.

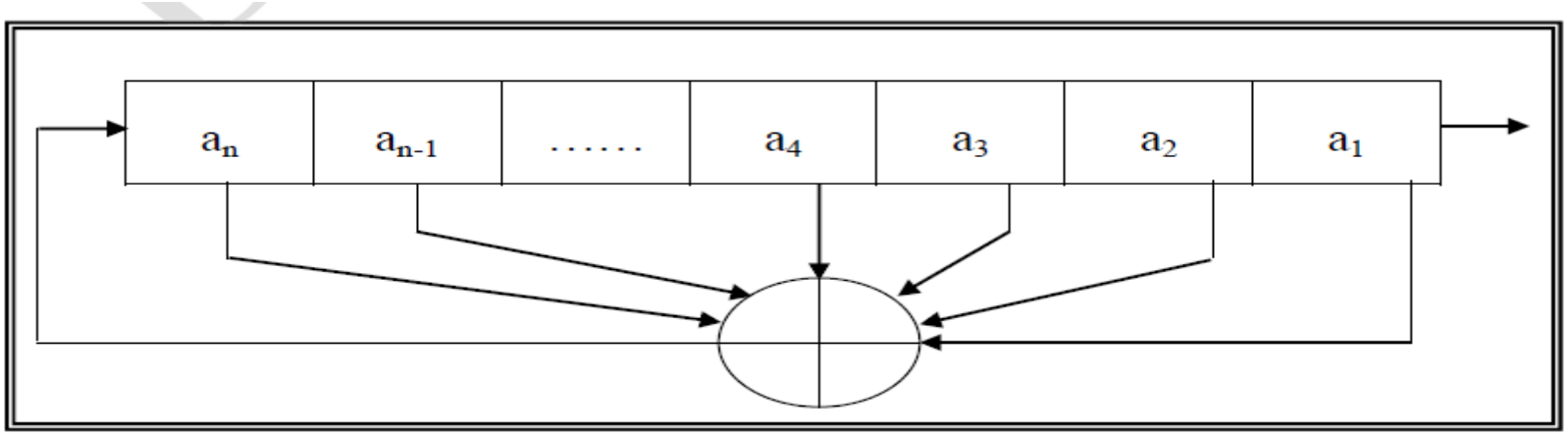

Figure (8) Linear Feedback Shift Register.

Engineering Concepts of Shift Register

For example, a 3-bit LFSR tapped at the first and third bit. If it is initialized with the value 111, it produces the following sequence:

1 1 1

0 1 1

1 0 1

0 1 0

0 0 1

1 0 0

1 1 0

The output sequence is the string of least significant bits: 1110100…

In order for a particular LFSR to be a **maximal-period LFSR**, the polynomial must be a primitive polynomial mod 2. The degree of the polynomial is the length of the shift register. Most practical stream-cipher designs center around LFSR. In the early days of electronics, Linear feedback shift registers (LFSRs) are used in many of the keystream generators that have been proposed in the literature. There are several reasons for this:

- LFSRs are well-suited to hardware implementation.
- They can produce sequences of large period.
- They can produce sequences with good statistical properties.
- Because of their structure, they can be readily analyzed using algebraic techniques.

Mathematical Model of LFSR's-Systems

LFSR Unit

Every LFSR's-system consists of collection of LFSR's, every one shifted alone in one time, as the nature of connection function, each LFSR produces independent sequence.

The LFSR unit depends on:

- LFSR's length.
- Connection function.
- The initial values of LFSR.

Two LFSR's are said to be similar if they have equal length and the same connection function, otherwise the called different. The single LFSR considered the smallest LFSR's-system.

Combining Function (CF) Unit

The CF, denoted by F_n, is a Boolean function (we focus on Boolean function defined on GF(2)) its inputs are the sequences generated from each LFSR. If $x_1,x_2,\ldots,x_n$ are input of F_n s.t. $x_i \in GF(2)$, i=1,2,…,n then:

$$F_n = a_0 \oplus \sum_{i=1}^{n} a_i x_i x_j \oplus \sum_{i,j} a_{ij} x_i \oplus \cdots \oplus a_{12\ldots n} \prod_{i=1}^{n} x_i$$

Where $a_0,a_i,a_{ij},\ldots,a_{12\ldots n} \in GF(2)$ are the coefficients for combination of LFSR's combined in Boolean function.

Mathematical Model of LFSR's-Systems

Example: if all the coefficients are zero's except a_i=1, ∀i, then:

$L_n(x_1,x_2,\ldots,x_n)= \sum_{i=1}^{n} x_i$ …(2)

This function is the linear function.

if all the coefficients are zero's except $a_{12\ldots n}$=1, then:

$P_n(x_1,x_2,\ldots,x_n)= \prod_{i=1}^{n} x_i$ …(3)

This function is the non-linear product function.

if all the coefficients are zero's except $a_{12}= a_{13}= a_{23}$=1, then:

$M_n(x_1,x_2,\ldots,x_n)=x_1x_2 \oplus x_1x_3 \oplus x_2x_3$ …(4)

This function is non-linear called majority function.

The CF depends on following elements:

- **Input sequences**: they are the sequences which are generated form LFSR's.
- **Output sequence**: It's the sequence which produced from mixing the input sequences of CF.

Definition (1): We called the function F_n **balance** function if p(z=0)=p(z=1)=0.5, where z is the output variable of CF and p(z) is the probability of the output z=0 or 1, otherwise it is not balance.

Definition (2): We called the function F_n **symmetric** function if the arrangement of the input sequences don't effect on the output sequence.

Mathematical Model of LFSR's-Systems

Boolean Table of CF

It's also called Truth Table (TT) of CF, it's a table represent the behavior of the Boolean function for all input possibilities. As usual, its consists of n+1 column, n are the inputs variables x_i of F_n and one for output of function F_n, and 2^n row, because for n inputs there are 2^n possible. The important benefit of TT is finding the output value of each combination of inputs. And since the feedback function is Boolean function then we can express this function as TT.

Example: The TT of functions mentioned in eqs(2), (3) and (4) can be expressed in table, use n=3.

Input			Output		
x_1	x_2	x_3	L_3	P_3	M_3
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	1	1	1

Mathematical Model of LFSR's-Systems

Remark : form above table we have:

- Since n=3 then there are 2^3=8 input possible.
- L_3 and M_3 are balance, but P_3 is not.
- All three functions are symmetric.

The other benefit of TT, when the output of the TT is known but the function is not then the logical expression of the function can be known from the TT. If x_i denotes the variable x in the input i and $X=(x_1,x_2,\ldots,x_n)$, then:

$h_j(X)= \prod_{i=1}^{n} a_i$, j=1,…,2^n, s.t.

$$a_i = \begin{cases} x_i \oplus 1, & x_i = 0 \\ x_i, & x_i = 1 \end{cases}$$

The values of a_i changes when j changes, and then sum and multiply (module 2), therefore the logical expression of combining function is:

$F_n(x_1,x_2,\ldots,x_n)= \sum_{i=1}^{2^n \oplus} h_j(X) b_j$ …(5)

Where b_j is the output of the function at row j.

Example: the logical expression of the function F_3

$F_3(x_1,x_2,x_3)=(x_1\oplus1)(x_2\oplus1)(x_3\oplus1)\oplus(x_1\oplus1)x_2x_3\oplus x_1(x_2\oplus1)x_3\oplus x_1x_2x_3= x_1x_2\oplus x_1\oplus x_2\oplus x_3\oplus1$

x_1	x_2	⋯	x_n	h(X)	F_n
0	0	⋯	0	$(x_1\oplus1)(x_2\oplus1)\cdots(x_n\oplus1)$	b_1
0	0	⋯	1	$(x_1\oplus1)(x_2\oplus1)\cdots x_n$	b_2
⋮	⋮	⋮	⋮	⋮	⋮
1	1	⋯	0	$x_1x_2\cdots(x_n\oplus1)$	b_{2^n-1}
1	1	⋯	1	$x_1x_2\cdots x_n$	b_{2^n}

Input			Output
x_1	x_2	x_3	F_3
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Basic Building-Blocks of Stream Ciphers

Secure Combination Generator Properties

One approach is to use n LFSRs in parallel; their outputs combined using an n-input binary CF. the output sequence of an LFSR based keystream generator (KG) should have the following properties:

- Large period.
- Large linear complexity.
- Good statistical randomness properties.
- Correlation immune.

It is emphasized that these properties are only **necessary** conditions for a KG to be considered cryptographically secure. Since mathematical proofs of security of such generators are not known, such generators can only be deemed **computationally secure** after having withstood sufficient public scrutiny. The LFSRs in KG may have known or secret connection polynomials. For known connections, the secret key generally consists of the initial contents of the component LFSRs. For secret connections, the secret key for the KG generally consists of both the initial contents and the connections.

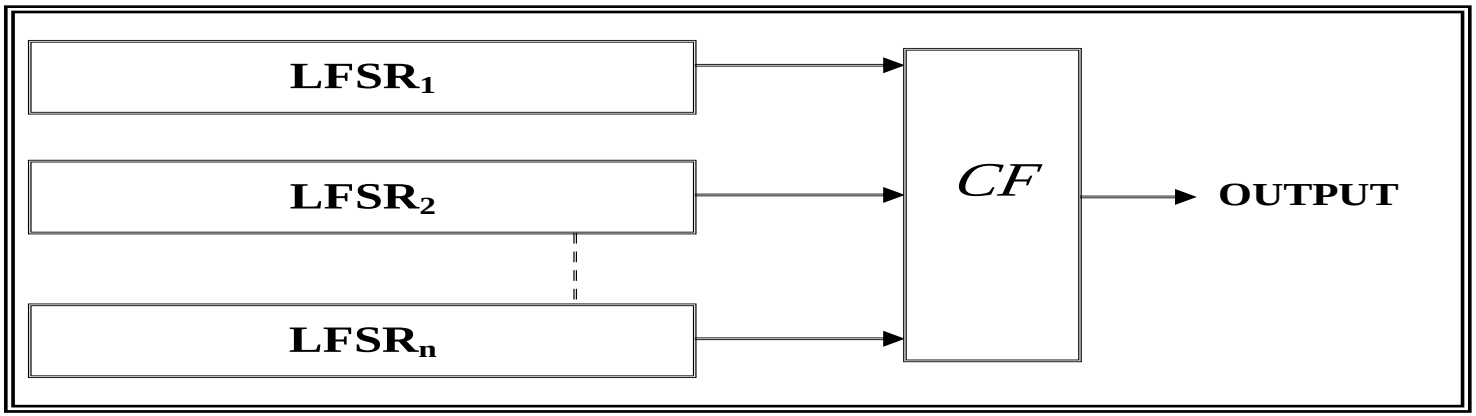

Figure (9) n-LFSR's Generator with Combining Function

Basic Building-Blocks of Stream Ciphers

Destroying the Linearity of LFSR's

Because LFSRs are inherently linear, one technique for removing the linearity is to feed the outputs of several parallel LFSRs into a non-linear Boolean function to form a **combination generator**. Various properties of such a CF are critical for ensuring the security of the resultant scheme, for example, in order to avoid correlation attacks.

Three general methodologies for destroying the linearity properties of LFSRs are discussed in this section:

- Using a nonlinear CF on the outputs of several LFSRs.
- Using a nonlinear filtering function on the contents of a single LFSR.
- Using the output of one (or more) LFSRs to control the clock of one (or more) other LFSRs.