Lecture Three

Stream Cipher and Shift Register

11. Basic Building-Blocks of Stream Ciphers

11.1 Secure Combination Generator Properties

One approach is to use n LFSRs in parallel; their outputs combined using an n-input binary **Boolean function** or **combining function** (CF) figure (9) shows the design of n-LFSR's generator with combining function.

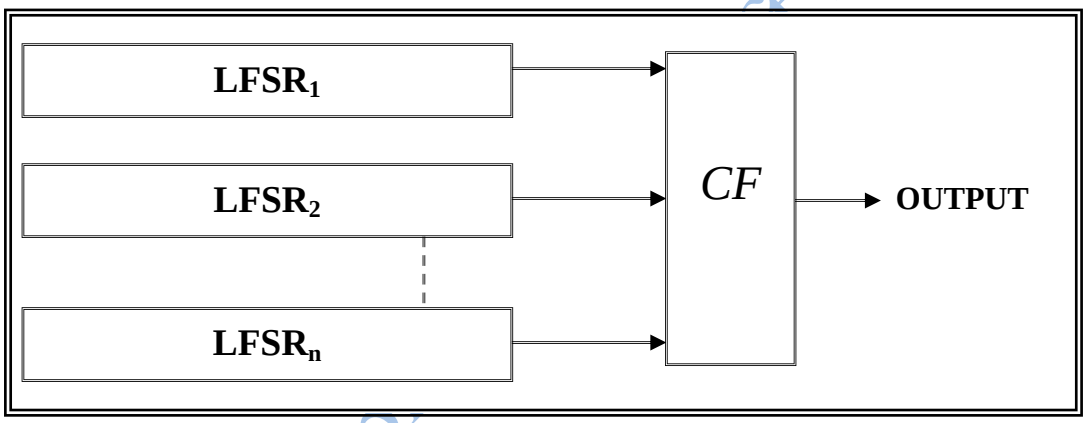

Figure (9) n-LFSR's Generator with Combining Function

For essentially all possible secret keys, the output sequence of an LFSR based keystream generator should have the following properties:

1. Large period.
2. Large linear complexity.
3. Good statistical randomness properties.
4. Correlation immune.

It is emphasized that these properties are only **necessary** conditions for a keystream generator 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 an LFSR-based keystream generator 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 keystream generator generally consists of both the initial contents and the connections.

The general conditions to designing a keystream generator using LFSR are mentioned in lecture five.

11.2 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 combining function 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:

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

12. Common Examples LFRS's-Systems

Some common examples of keystream generators are introduced.

1. Linear Generator

The **Linear generator** is defined by n-maximum-length LFSRs whose lengths $r_1, r_2,\ldots, r_n$, where $n \in Z^+$ are pair wise relatively prime, with XOR combining function:

$$F(x_1,x_2,..,x_n) = x_1 \oplus x_2 \oplus \ldots \oplus x_n \quad \ldots (11.1)$$

This generator considered weak, despite of his good randomness, because of his weak linear complexity. Figure (10) shows the Linear generator.

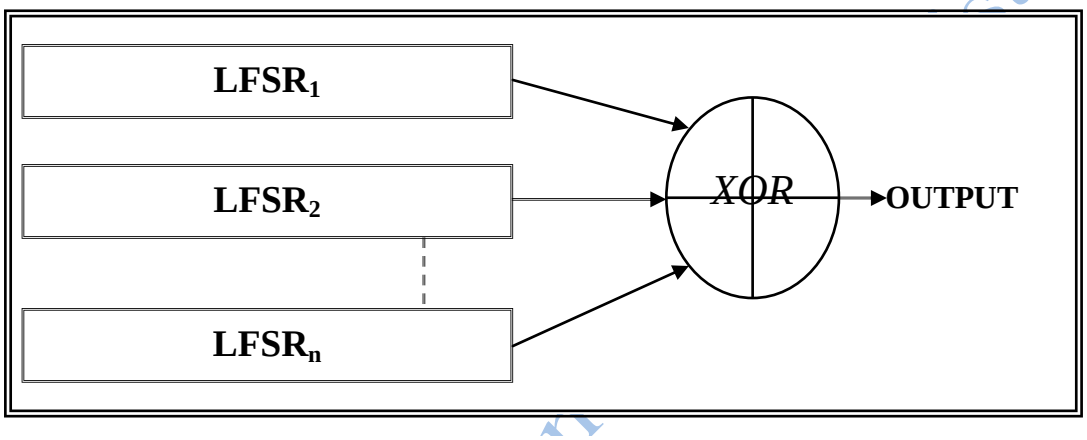

Figure (10) n-Linear Generator.

2. Product Generator

The **Product generator** is defined by n-maximum-length LFSRs whose lengths $r_1, r_2,\ldots, r_n$, where $n \in Z^+$ are pair wise relatively prime, with AND combining function:

$$F(x_1,x_2,..,x_n) = x_1 \bullet x_2 \bullet \ldots \bullet x_n = \prod_{i=1}^{n} x_i \quad \ldots (11.2)$$

This generator considered weak, despite of his good linear complexity, because of his weak randomness. Figure (11) shows the Product generator.

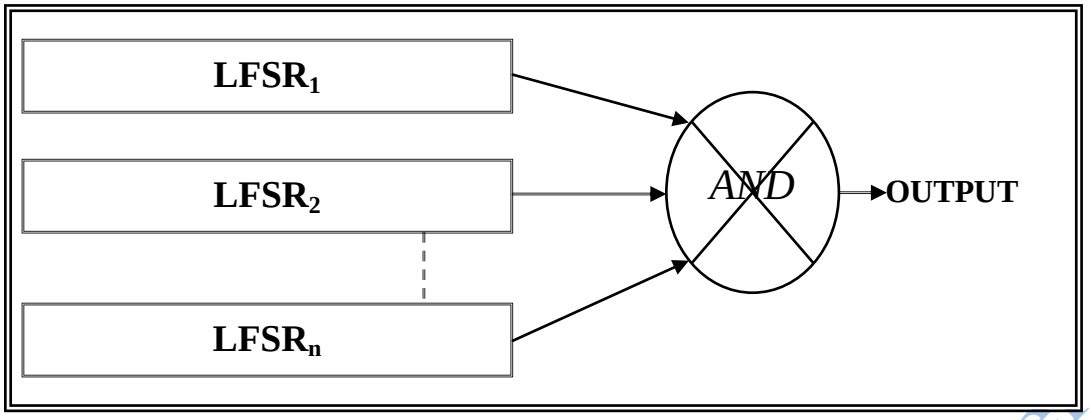

Figure (11) n-Product Generator.

3. Brüer Generator

As usual, the Brüer generator consists of odd number of LFSR's (here it taken 3 LFSR's), whose lengths r_1, r_2, r_3 are pair wise relatively prime, with nonlinear combining function (also called majority function):

$$F(x_1,x_2,x_3)=x_1x_2\oplus x_1x_3\oplus x_2x_3 \quad \ldots(11.3)$$

The keystream generated has period $(2^{r_1}-1)(2^{r_2}-1)(2^{r_3}-1)$ and linear complexity $L=r_1r_2+r_1r_3+r_2r_3$. Figure (12) shows the Brüer generator.

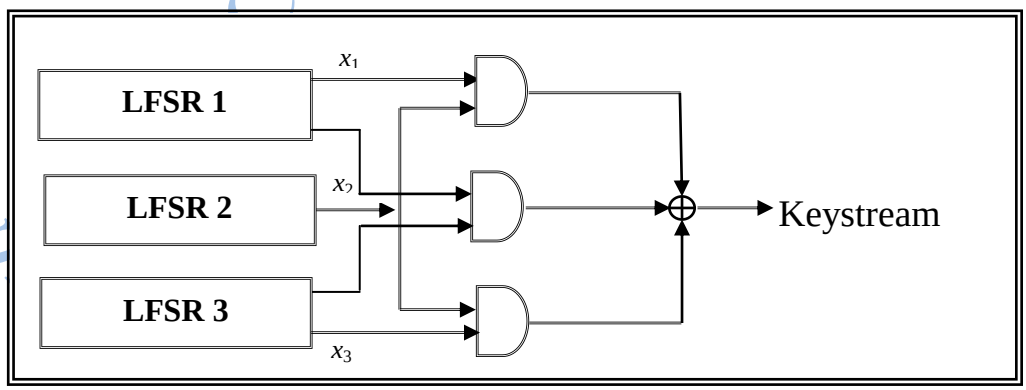

Figure (12) The Brüer generator