Stream Ciphers (contd.)

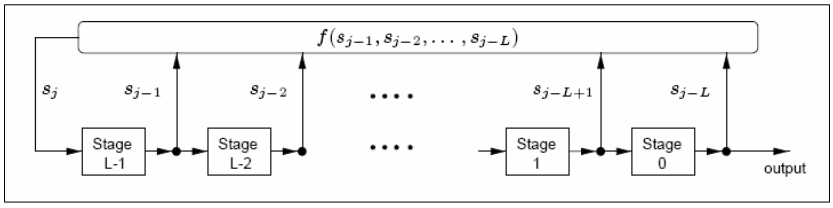
Debdeep Mukhopadhyay

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
INDIA -721302

Objectives

- **Non-linear feedback shift registers**
- **Stream ciphers using LFSRs:**
  - **Non-linear combination generators**
  - **Non-linear filter generators**
  - **Clock controlled generators**
  - **Other Stream Ciphers**

Non-linear feedback shift registers



- **A Feedback Shift Register (FSR) is non-singular iff for all possible initial states every output sequence of the FSR is periodic.**

de Bruijn Sequence

An FSR with feedback function $f(s_{j-1}, s_{j-2}, ..., s_{j-L})$ is non-singular iff f is of the form:

$$f = s_{j-L} \oplus g(s_{j-1}, s_{j-2}, ..., s_{j-L+1})$$

for some Boolean function g.

The period of a non-singular FSR with length L is at most 2^L.

If the period of the output sequence for any initial state of a non-singular FSR of length L is 2^L, then the FSR is called a *de Bruijn* FSR, and the output sequence is called a *de Bruijn sequence.*

Example

$$f(x_1, x_2, x_3) = 1 \oplus x_2 \oplus x_3 \oplus x_1 x_2$$

t	x_1	x_2	x_3
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1

t	x_1	x_2	x_3
4	0	1	1
5	1	0	1
6	0	1	0
3	0	0	1

Converting a maximal length LFSR to a de-Bruijn FSR

Let R_1 be a maximum length LFSR of length L with linear feedback function:

$$f(s_{j-1}, s_{j-2}, ..., s_{j-L}).$$

Then the FSR R_2 with feedback function:

$$g(s_{j-1}, s_{j-2}, ..., s_{j-L}) = f \oplus \bar{s}_{j-1}, \bar{s}_{j-2}, ..., \bar{s}_{j-L+1}$$

is a *de Bruijn* FSR.

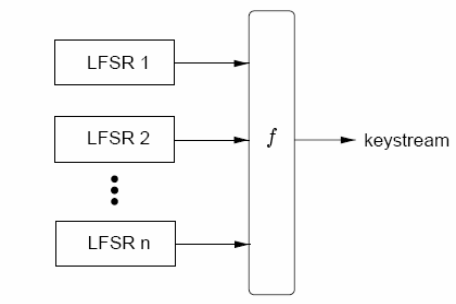
Stream Ciphers based on LFSR

- **LFSRs are popular key stream generators:**
  - **well suited for hardware implementations**
  - **large period**
  - **good statistical properties**
- **However the key stream is predictable**
  - ***the connection polynomial of an LFSR with linear complexity L can be efficiently computed from a sub-sequence of length 2L or more***
  - ***the sub-sequence can be ascertained by a known plain-text attack***

Three LFSR based methods

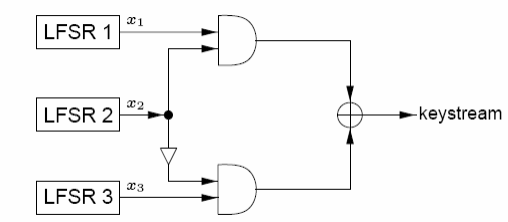
- **using a non-linear combining function on the outputs of several LFSRs**
- **using a non-linear filtering function on the contents of a single LFSR**
- **using the output of one (or more) LFSR to control the clock of one (or more) LFSR**

Non-linear combination generators



- **f is a non-linear combining function**
- **Suppose that n maximum length LFSRs, whose lengths L_1, L_2, …,L_n are pairwise distinct and greater than 2, are combined by a non-linear function $f(x_1,x_2,\ldots,x_n)$, which is the ANF form. Then the linear complexity of the key stream is $f(L_1, L_2,\ldots,L_n)$, where the xors are replaced by integer additions.**

Example: the Geffe generator



- **3 maximum length LFSRs whose lengths L_1, L_2 and L_3 are pairwise relatively prime.**
- **Period=$(2^{L1}-1)(2^{L2}-1)(2^{L3}-1)$**
- **Linear Complexity=$L_1L_2+L_2L_3+L_3$**

Correlation Attacks

- **The Geffe generator is cryptographically weak, because information about the states of LFSR 1 and LFSR 3 leaks into the output sequence.**

$$\Pr(z(t)=x_1(t)) = \Pr(x_2(t)=1) + \Pr(x_2(t)=0)\Pr(x_3(t)=x_1(t))$$
$$= \frac{1}{2} + \frac{1}{2}\frac{1}{2} = \frac{3}{4}$$

Similarly, $\Pr(z(t)=x_3(t)) = \frac{3}{4}$

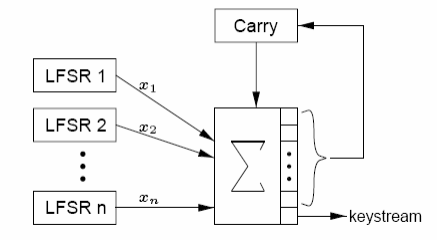
Correlation attacks

- **Consider n maximum length LFSRs $R_1,R_2,\ldots,R_n$ with lengths $L_1,L_2,\ldots,L_n$**
- **Number of keys=$(2^{L1}-1)(2^{L2}-1)\ldots(2^{Ln}-1)$**
- **Suppose that there is a correlation between the keystream and the output of R_1 with probability $p>1/2$**
  - **guess the initial state of R_1**
  - **Compute the number of coincidences between the keystream and all possible shifts of the output sequence of R1, until the probability is more than p.**
  - **Number of trials=$(2^{L1}-1)$**
  - **Since the initial states of the LFSRs can be known independently, total number of trials=$\Sigma(2^{Li}-1)$**

Correlation Immunity

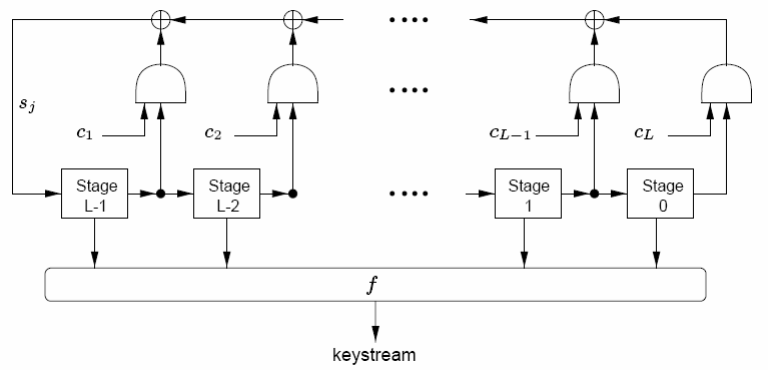
- **Let X_1, …,X_n be independent binary variables, each taking values 0 or 1 with probability ½**
- **A Boolean function f is m^{th} order correlation immune if for each subset of m random variables X_{i1}, X_{i2},…, X_{im}, the random variable $Z=f(X_1,…,X_n)$ is statistically independent of the random vector (X_{i1}, X_{i2},…, X_{im})**

Summation Generator



- **The lengths L_1,L_2,…,L_n of the n LFSRs are pairwise prime.**
- **Period of the key-stream=$\Pi(2^{Li}-1)$, while its linear complexity is close to this number.**

Non-linear filter generators

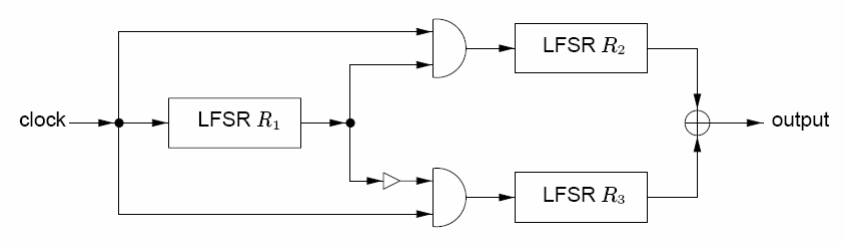


Clock controlled generators

Alternating Step Generator:

- **A control LFSR R_1 is used to selectively step two other LFSRs, R_2 and R_3.**
- **Output sequence is the XOR of R_2 and R_3.**
- **Algorithm:**
  - **Register R_1 is clocked.**
  - **If output of R_1 is 1, then R_2 is clocked, R_3 is not clocked but the previous output is repeated.**
  - **If output of R_1 is 0, then R_3 is clocked, R_2 is not clocked but the previous output is repeated.**

Example



- **If R_1 produces a de Bruijn sequence, the alternating step generator has high period, high linear complexity, and have good statistical properties.**

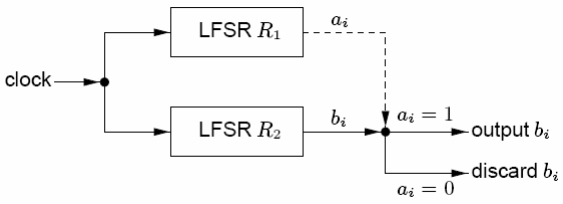
Example (contd.)

- **R_1=<3,1+D^2+D^3>, R_2=<4,1+D^3+D^4>, R_3=<5,1+D+D^3+D^4+D^5>**
- **Suppose initial states of R_1=[001], R_2=[1011], R_3=[01001]**
- **Output sequences:**
  - **R_1: 1001011**
  - **R_2:1101 0111 1000 100**
  - **R_3:1001 0101 1000 0111 0011 0111 1101 000**
  - **z: 1011 1010 1010 0001 0111 1011 0001 110…**

Shrinking generator

- **a control LFSR R_1 is used to control the output of a second LFSR R_2**
- **Register R_1 and R_2 are clocked**
- **If the output of R_1 is 1, the output bit of R_2 forms part of the key stream**
- **If the output of R_1 is 0, the output of R_2 is discarded.**

Example



- **R_1 and R_2 are maximum length LFSRs**
- **L_1 and L_2 are mutually co-prime, and if the connection polynomials are unknown then the security level is $\approx 2^{2l}$, where $L_1 \approx l$, $L_2 \approx l$**
- **thus keeping l=64, the SG should be quite strong.**

Example (contd.)

- **R_1=<3,1+D+D^3>, R_2=<5,1+D^3+D^5>**
- **Suppose initial states of R_1=[100], R_2=[00101]**
- **Output sequences:**
  - **R_1: 0011101**
  - **R_2:1010 0001 0010 1100 1111 1000 1101 110**
  - **x: 1000 0101 1111 1011 10…**

Modern Stream Ciphers

- **Several proposals in the Estream Website**
- **There are hardware and software candidates**
- **Search for standard Stream Ciphers**
- **New attack techniques have been developed, like algebraic attacks, cube attacks.**
- **Stream cipher design have become all the more challenging.**

Points to Ponder!

- **A self-shrinking generator (SSG) uses only one maximum length LFSR R. The output sequence of R is partitioned into pairs of bits.**
- **The SSG outputs:**
  - **0 if the pair is 10**
  - **1 if the pair is 11**
  - **00 and 01 pairs are dropped**
- **A SSG can be implemented as a shrinking generator and vice-versa. Can you work out?**

Further Reading

- **A. Menezes, P. Van Oorschot, Scott Vanstone, "Handbook of Applied Cryptography" (Available online)**

Next Days Topic

- **Pseudorandomness**