*Guang Gong*
*Dept. of Electrical and Computer Engineering*
*University of Waterloo*

University of Waterloo

# Design of Stream Ciphers

**Short Course of Four Lectures:**

- **Historic Developments of Pseudo-Random Sequence Generators (PRSG) and Stream Ciphers**
- **Linear Feedback Shift Register Sequences**
- **Randomness Measurements**
- **Design of PRSGs Towards Large Linear Span**
- **Examples of Stream Ciphers in Practice**

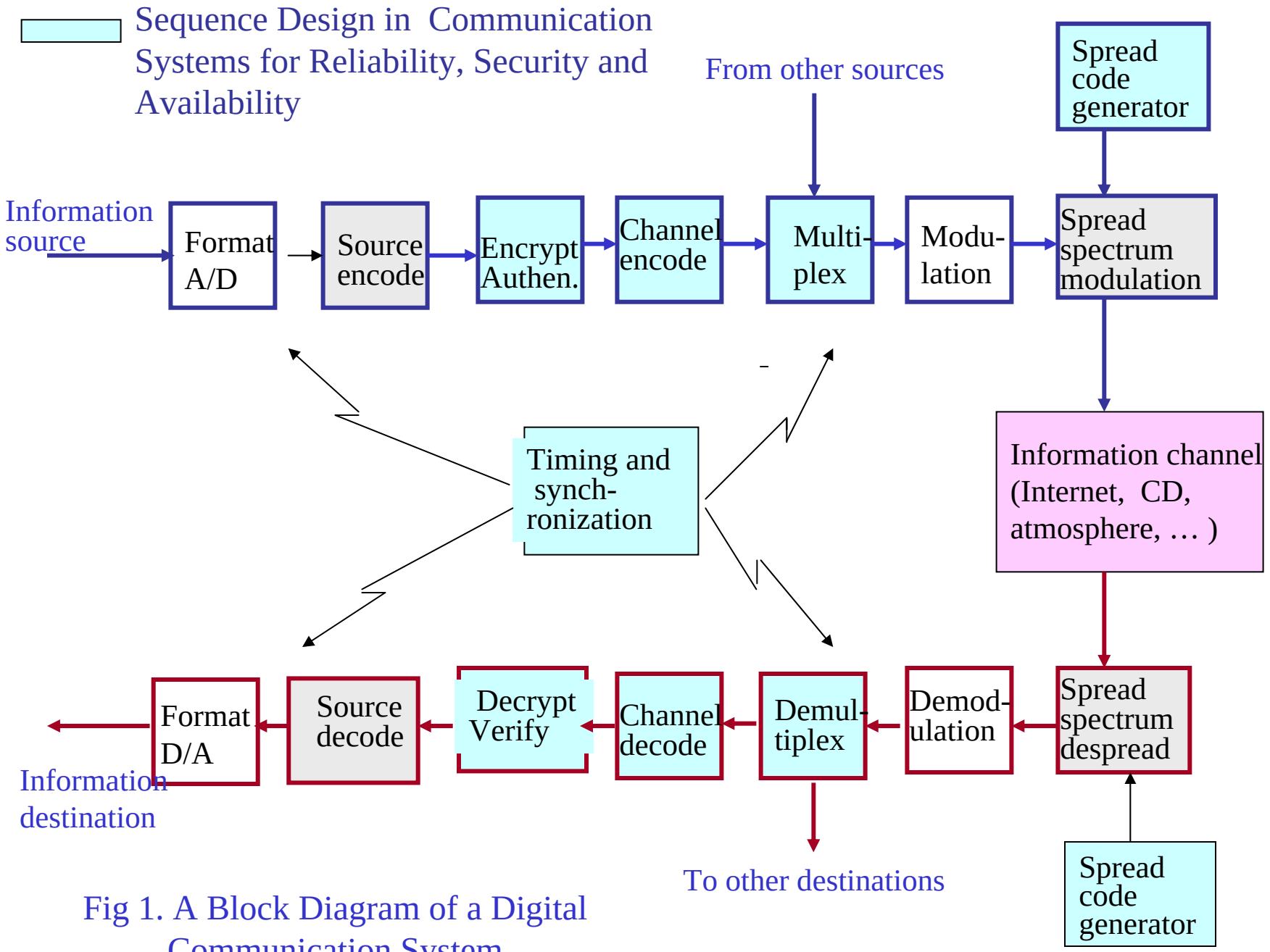
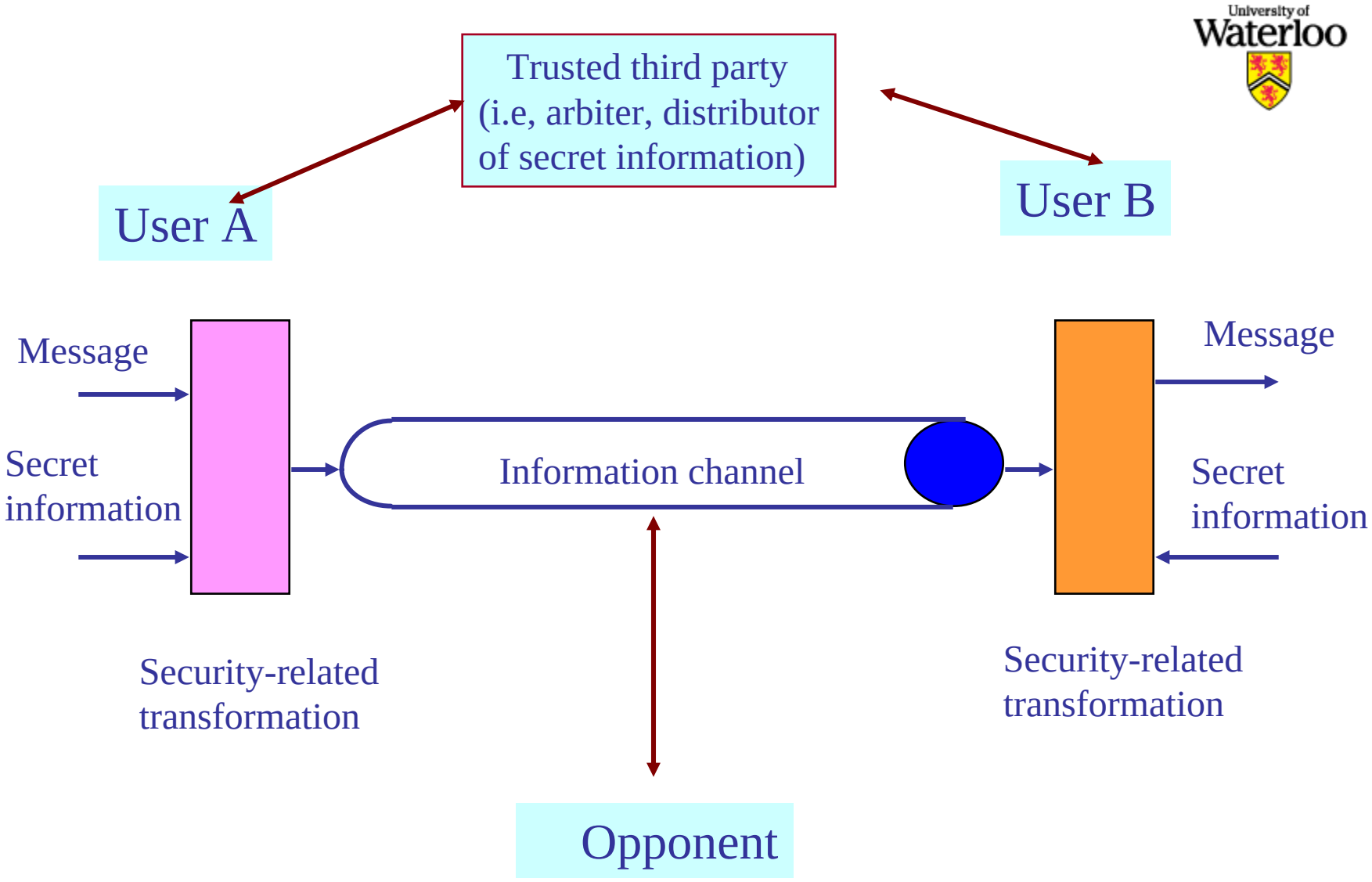



Fig 1. A Block Diagram of a Digital Communication System

# Model for Secure Communications

**Three components of secure communications:**

- Communication principals
- Trusted third party (or authority)
- Opponents (attackers)




**Model for Secure Communications**

Insecure information channel

Sender

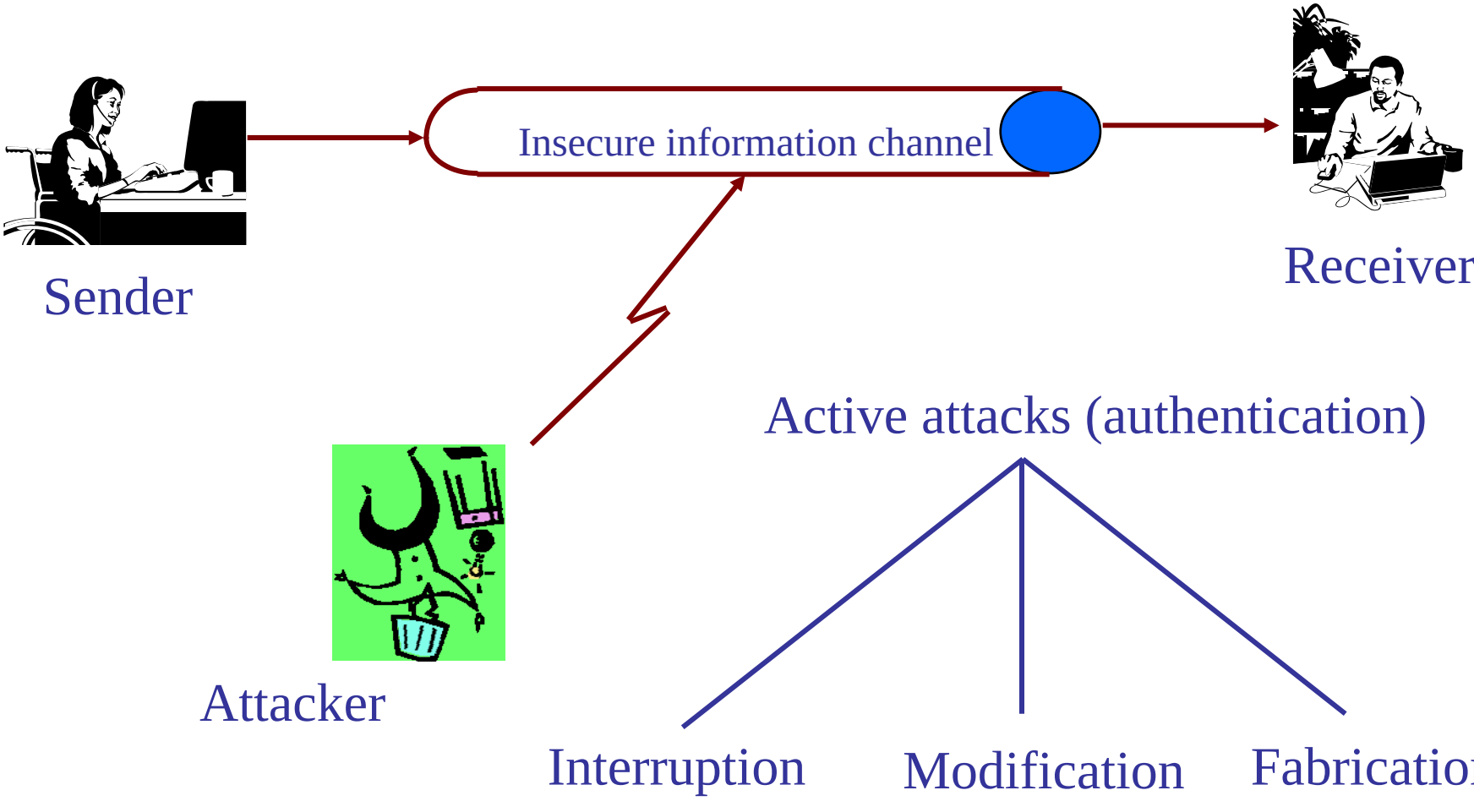
Receiver

Attacker

Passive attacks

Interception (confidentiality, privacy)

**Passive attacks**




**Active attacks**

# Historical Developments of Pseudo-Random Sequences

University of Waterloo

- Period of pre-application (before 1948)
- Golden period of m-sequences (1948-1969):

  Applications:

  - Stream cipher (e.g. voice encryption)
  - Radar and sonar distance range, synchronization
- Period of non-linear generators (1969 - present):

  Applications:

  - Stream ciphers and light cryptography
  - Code division multiplex access (CDMA)
  - Error correction code
  - Distance range, synchronization, identification code, and hardware testing

# Golden period of m-sequences (1948-1969)

- Shannon's Result (1948): One-time-pad is unbreakable

- Berlekamp-Massey algorithm attack (1969)

  LFSR generates *m*-sequences

  = Maximal length sequences

  = Pseudo-noise (PN) sequences

  = PSG (1948-1969)

# Period of Non-Linear Generators for Good Randomness (Or Unpredictabilty)

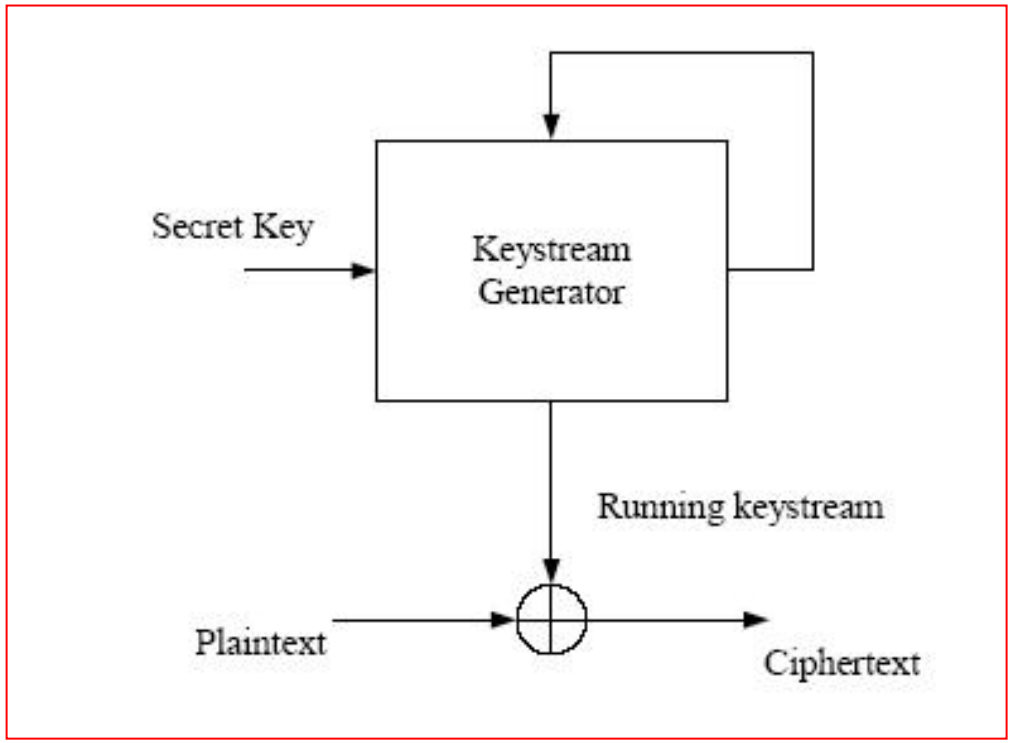
(a) Design towards 2-Level Auto-Correlation and Low Correlation

(b) Design towards Large Linear Span

LFSR as Basic Blocks

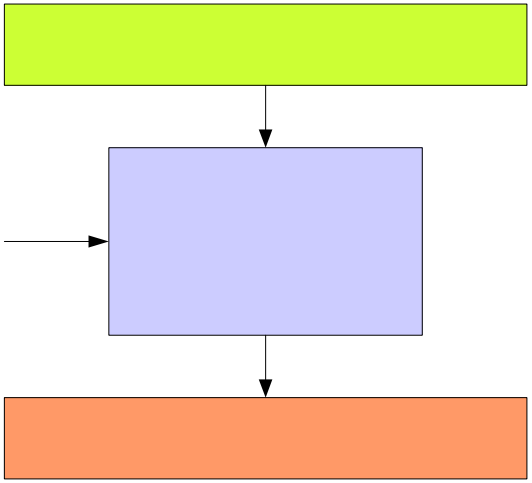
# Historical Progress of Stream Ciphers

University of Waterloo

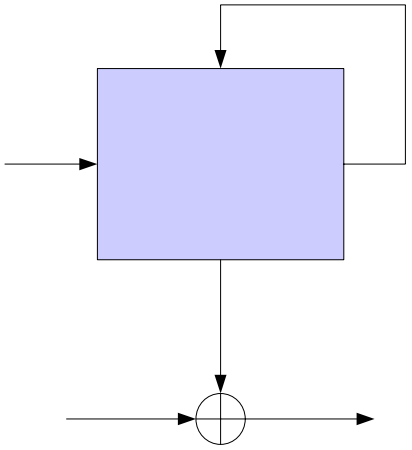



General Model of a Stream Cipher

# Symmetric Key Encryption



Block Cipher



Stream Cipher

# Block Ciphers vs Stream Ciphers

## Block Ciphers

- No theoretical proof for security, but it can satisfy some sufficient conditions in Shannon's Theory such as confusion and diffusion among messages, keys and ciphertexts, and no synchronous problem.
- Historically more resistant to cryptanalytic attacks
- Hardware: Moderate speed and complexity
- Software: Moderate speed

## Stream Ciphers

- Shannon in his milestone work showed that one-time-pad is unconditional secure – leads to a fantasy to search for secure key stream generators. Problem: hard to resynchronization between transmitters and receivers
- Most of the proposed stream ciphers suffered effective cryptanalytic attacks
- Hardware: High speed and low complexity (*ideal for constrained platforms: PDA's, sensors, etc.*)
- Software: High speed

# Efforts for Standardization of Steam Ciphers

➢In 2001-2004, a new initiative known as the New European Schemes for Signatures, Integrity, and Encryption (NESSIE) announced its call for cryptographic primitives including stream ciphers. The latter could become new encryption algorithms for the 3G and 4G systems. However, it did not make any recommendations for the submitted candidates.

➢ ECRYPT is a Network of Excellence within the Information Societies Technology (IST) Program of the European Commission. The ECRYPT Stream Cipher Project is a multi-year (2004-2008) project to identify new stream ciphers that might become suitable for widespread adoption. There are 34 submissions.

The first phase was finished in the middle of March 2006, and WG has been advanced to the second phase of the evaluation. The final report will be made in January 2008. WG stream cipher is the only one which possesses the required pseudo-randomness properties.

# 2. Linear Feedback Shift Registers (LFSR)

- **Feedback Shift Registers (FSR)**
- **Characteristic Polynomials and Periods of LFSR**

# 1.1 Feedback Shift Registers (FSR)

## A. Basic Concepts and Examples: Binary Case

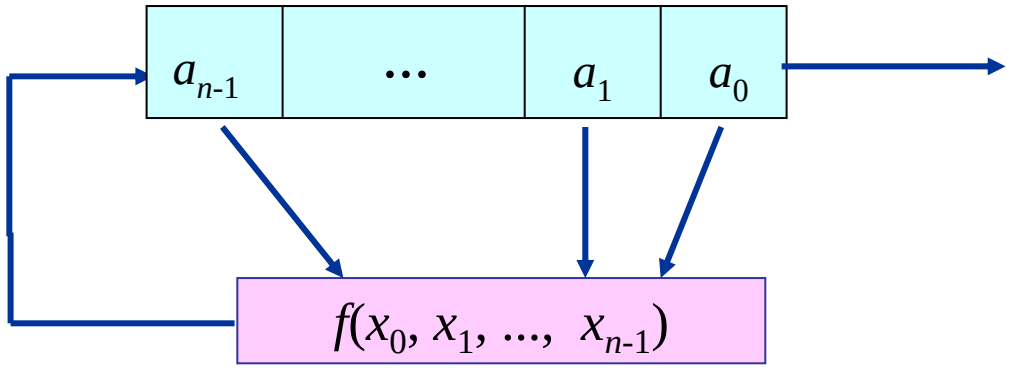



Fig 1. A Block Diagram of an FSR

# Three Components

(1) $n$-stage shift register : $n$ 2-state storage units

(2) Initial state ($a_0$, $a_1$, ..., $a_{n-1}$)

(3) Feedback function : $f(x_0, ..., x_n)$ is a boolean function in $n$ variables:

$$f(x_0, \cdots, x_{n-1}) = \sum_{\{i_1, \cdots, i_s\} \subset \{0, \cdots, n-1\}} a_{i_1 \cdots i_s} x_{i_1} \cdots x_{i_s}, \quad a_{i_1 \cdots i_s} \in \{0, 1\}$$

There are $2^{2^n}$ Boolean functions in $n$ variables.

**How does it work?**

At each clock pulse: the state of each memory stage is shifted to the next stage in line, *i.e.*, there is a transition from one state to next.

For example, the next state of Fig. 1 is


where

$$a_n = f(a_0, a_1, \ldots, a_{n-1})$$

and the device outputs one bit $a_0$.

So, we have a sequence

$$\{a_i\} = a_0, a_1, \ldots, a_{n-1}, a_n, \ldots$$

where the recursive relation is given by

$$a_{n+k} = f(a_k, a_{k+1}, \ldots, a_{k+n-1}) \text{ for } k = 0, 1, \ldots$$

The sequence $\{a_i\}$ is said to be an FSR sequence and a vector

$$\mathbf{s}_i = (a_i, a_{i+1}, ..., a_{i+n-1}),\ i = 0, 1, \ldots,$$

is called the *ith state* of the FSR or the sequence.

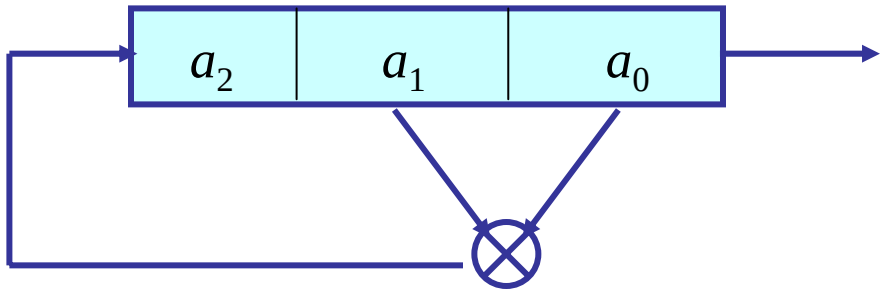
If $f(x_0, x_1, ..., x_{n-1})$ is linear, *i.e.*,

$$f(x_0, x_1, ..., x_{n-1}) = c_0x_0 + c_1x_1 + \ ... \ + c_{n-1}x_{n-1},\ c_i \in \{0,1\},$$

then $\underline{\boldsymbol{a}} = \{a_i\}$ is called an LFSR sequence. Otherwise, an nonlinear FSR (NLFSR) sequence.

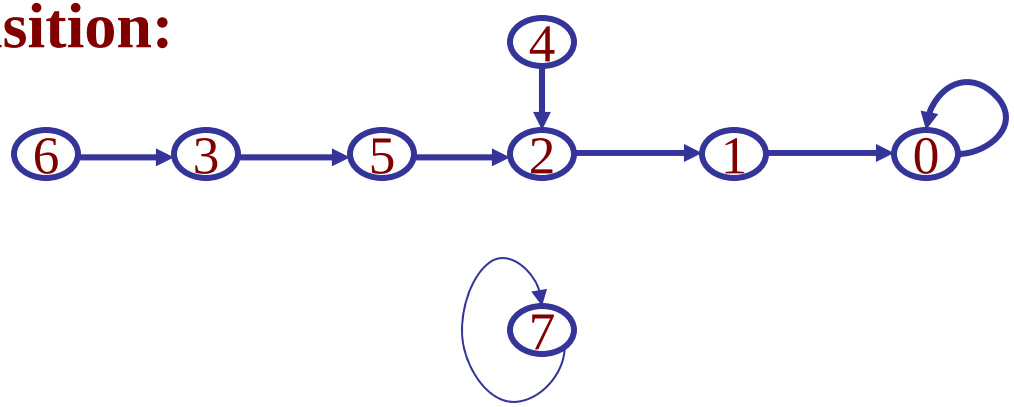
**Example 1.** A 3-stage NFSR with a feedback function

$$f(x_0, x_1, x_2) = x_0 x_1$$




| | Current state | Next state |
|---|---|---|
| | $x_2$ $x_1$ $x_0$ | $x_2$ $x_1$ $x_0$ |
| 0 | 0 0 0 | 0 0 0 |
| 1 | 0 0 1 | 0 0 0 |
| 2 | 0 1 0 | 0 0 1 |
| 3 | 0 1 1 | 1 0 1 |
| 4 | 1 0 0 | 0 1 0 |
| 5 | 1 0 1 | 0 1 0 |
| 6 | 1 1 0 | 0 1 1 |
| 7 | 1 1 1 | 1 1 1 |

**State transition:**




**Output sequences for different initial states:**

$$\underline{100}000\cdots \quad (a_0, a_1, a_2) = (1,0,0)$$

$$\underline{011}01000\cdots \quad (a_0, a_1, a_2) = (0,1,1)$$

$$\underline{001}000\cdots \quad (a_0, a_1, a_2) = (0,0,1)$$

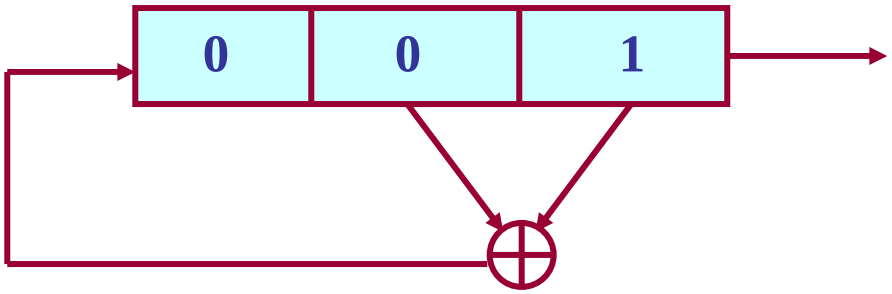
$$\underline{111}111\cdots \quad (a_0, a_1, a_2) = (1,1,1)$$

**Recursive relation:** $a_{3+k} = a_k a_{1+k}, \ k = 0, 1, \cdots$

**Example 2.** A 3-stage LFSR with a feedback function

$$f(x_0, x_1, x_2) = x_0 + x_1$$

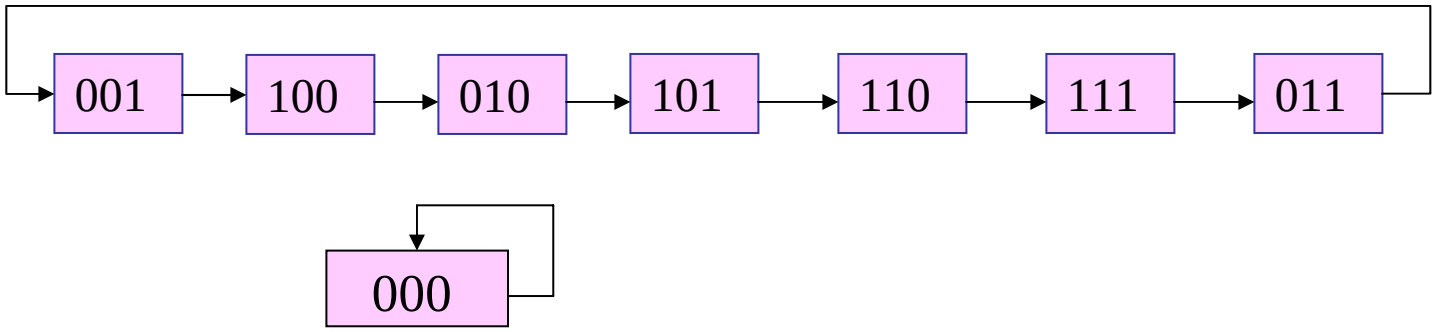
(1) Implementation

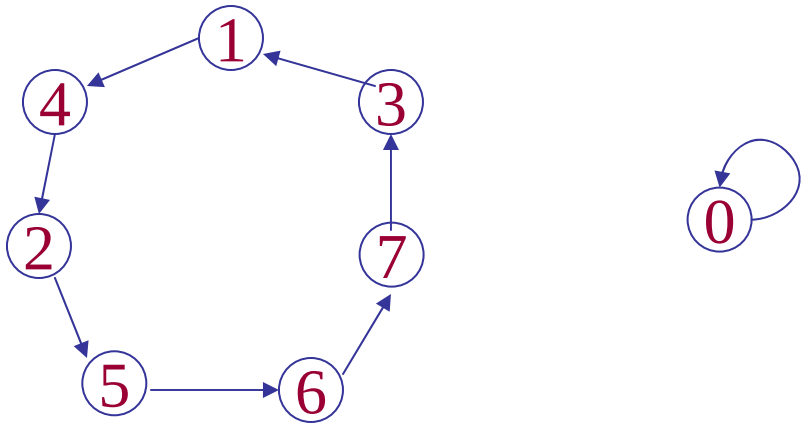



(2) Recursive relation:

$$a_{3+k} = a_{1+k} + a_k, k = 0, 1, \cdots$$

(3) State Diagram





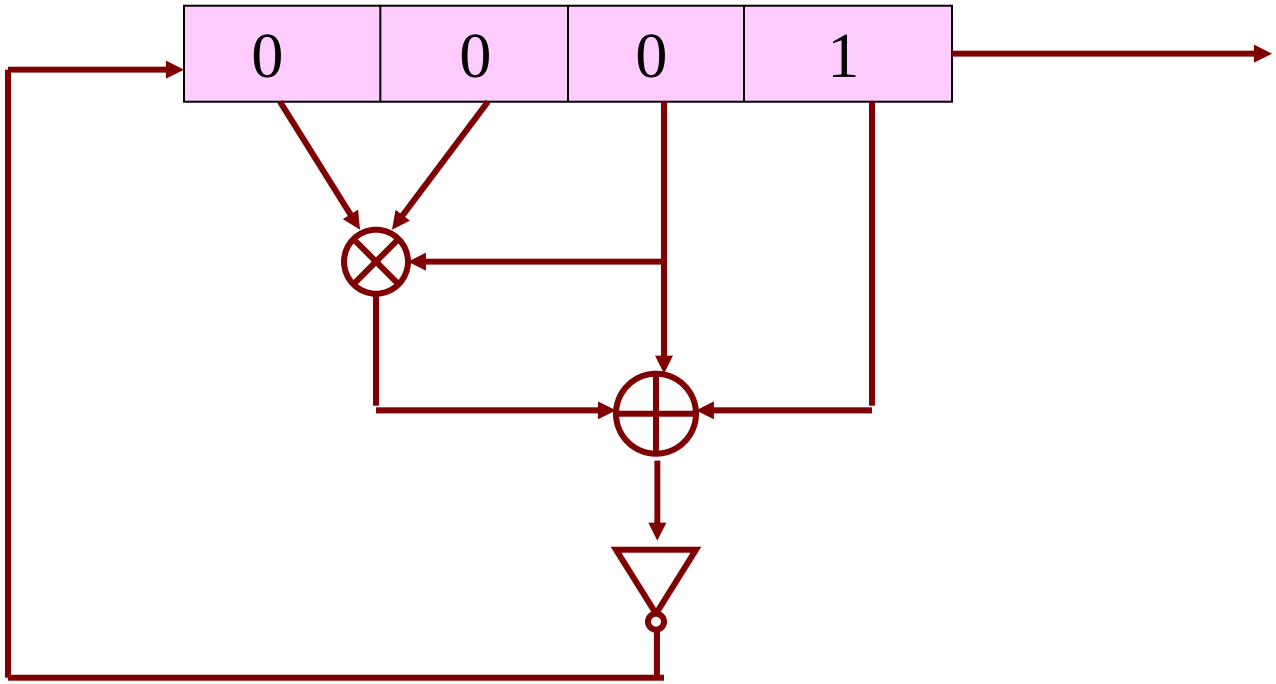

(4) Outputs with different initial states:

| Initial state: $(a_0, a_1, a_2)$ | Output sequence: |
|---|---|
| (1, 0, 0): | 10010111001011… |
| (1, 1, 1): | 11100101110010… |
| (0, 0 0): | 0000000… |

**Example 3.** Let the feedback function be given by

$$f(x_0, x_1, x_2, x_3) = 1 + x_0 + x_1 + x_1 x_2 x_3$$

A 4-stage FSR:



# Truth Table

$g(x_0, x_1, x_2, x_3) = x_0 + x_1 + x_1x_2x_3$ and $f = g + 1$

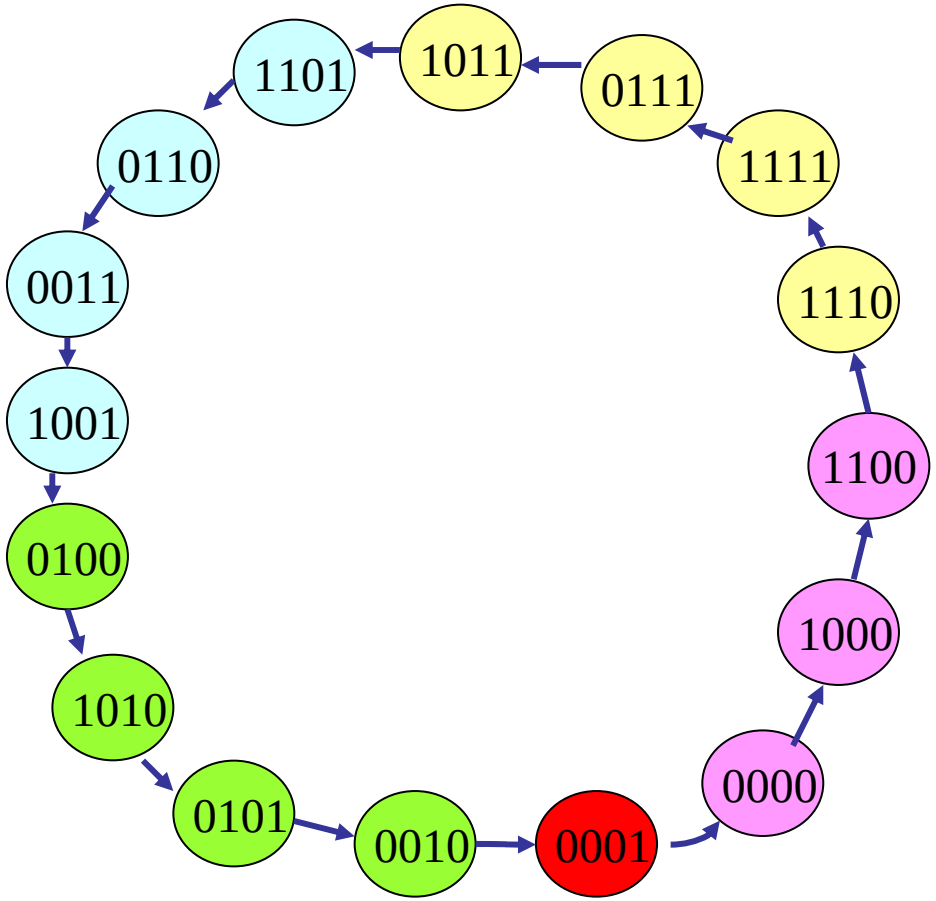
| $x_0$ | $x_1$ | $x_2$ | $x_3$ | $g$ | $f$ | $x_0$ | $x_1$ | $x_2$ | $x_3$ | $g$ | $f$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

The output sequence:

1000 011110101100 1000011110101100…

which has period 16.

A de Bruijn sequence is an output sequence of $n$-stage NLFSR with period $2^n$.




**State Diagram**

# B. Formal definition of *q*-ary FSR sequences

## An Abstract model: Let

- $F = GF(q)$, a finite field of order $q$, where $q$ is a prime or a power of prime.
- $f(x_0, x_1, ...., x_{n-1})$ be a function in n variables defined as

$$f(x_0,\cdots,x_{n-1}) = \sum_{\{i_1,\cdots,i_s\}\subset\{0,\cdots,n-1\}} c_{i_1\cdots i_s} x_{i_1}^{e_{i_1}} \cdots x_{i_s}^{e_{i_s}},\ c_{i_1\cdots i_s} \in F.$$

where $e_{i_j}$ are positive integers with $1 \le e_{i_j} < q, 0 \le s \le n.$

- $\underline{\mathbf{a}} = \{a_i\}$, $a_i \in F$ whose elements are given by

$$a_{n+k} = f(a_k, a_{k+1}, ...., a_{k+n-1}),\ k = 0, 1, ... .$$

Then

- $\underline{\mathbf{a}}$ is said to be a $q$-ary FSR sequence over $F$,
- $(a_0, a_1, ..., a_{n-1})$ is called an initial state of $\underline{\mathbf{a}}$
- $f(x_0, x_1, ..., x_{n-1})$ is called the feedback function of $\underline{\mathbf{a}}$.

If an feedback function $f(x_0, ..., x_{n-1})$ is linear, *i.e.*,

$$f(x_0, x_1, ...., x_{n-1}) = c_0x_0 + c_1x_1 + ... + c_{n-1}x_{n-1}, \quad c_i \in F,$$

then the recursive relation becomes a linear recursive relation:

$$a_{n+k} = \sum_{i=0}^{n-1} c_i a_{i+k}, \quad k = 0, 1, \cdots$$

So an LFSR sequence $\{a_i\}$ is also called a linear recursive sequence over $F$.

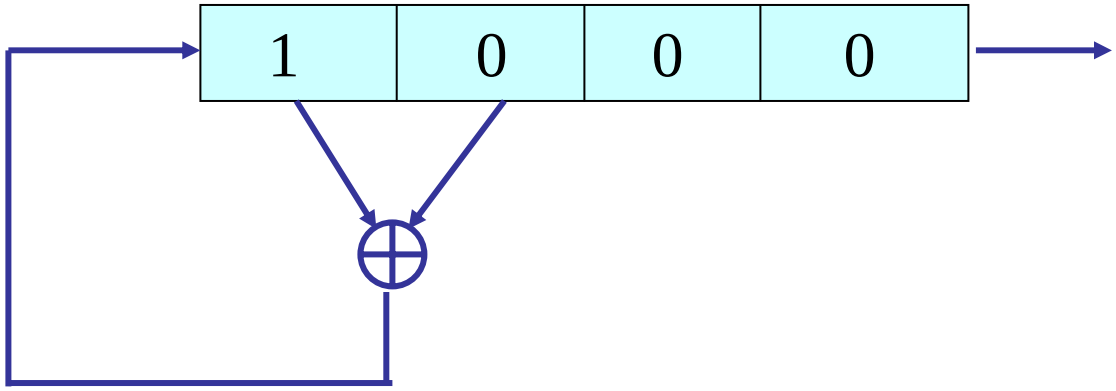
## C. Periodic Property

**Def.** Let $\underline{\mathbf{a}} = \{a_i\}$, $a_i \in F$. If there exists integer $r > 0$ and $u \geq 0$ such that

$$a_{i+r} = a_i, \text{ for all } i \geq u,$$

then the sequence is said to be ultimately periodic and $r$ is called a period of the sequence. The smallest integer satisfies by the above identity is called a least period of the sequence.

**Example 4.** Let $F = GF(2)$




4 - stage LFSR with $f(x_0, x_1, x_2, x_3) = x_2 + x_3$.

Output: 00011011011…

which is ultimately periodic with $u = 2$ and $r = 3$.

- Any output sequence of an $n$-stage FSR over $F$ is ultimately periodic with period $r \leq q^n$. In particular, if $q = 2$, then period $r \leq 2^n$.

- If the feedback function is linear, then any output of the LFSR is ultimately periodic with period $r \leq q^n - 1$. In particular, if $q = 2$, then period $r \leq 2^n - 1$. (Why?)

**Question**: How can one determine the periodic property of the output sequences of FSR or LFSR? In other words, under which conditions, does the state diagram has no branches?

## 1. 2 Characteristic Polynomials and Periods of LSFR

Let **a** be a LFSR sequence with a feedback function

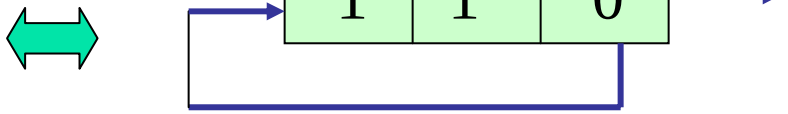
$$g(x_0, x_1, \cdots, x_{n-1}) = \sum_{i=0}^{n-1} c_i x_i, \text{ then } a_{n+k} = \sum_{i=0}^{n-1} c_i a_{i+k}, \quad k = 0,1,\cdots$$

Let $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_1 x - c_0$

Then $f(x)$ is said to be a characteristic polynomial of **a** or the LFSR defined by $g(x_0, x_1, \ldots, x_{n-1})$, **a**, generated by $f(x)$, and the reciprocal polynomial of $f(x)$, a feedback polynomial.

**Example 5 .** Let $q = 2$ and

$$f(x) = x^3+1.$$



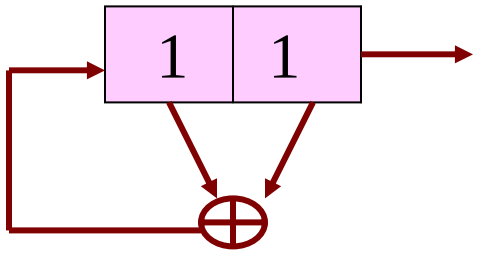
**a** = 011011011...




However, **a** can also be generated by an LFSR with a characteristic polynomial:

$$m(x) = x^2 + x + 1$$

**a** = 011011011...




## A. The minimal polynomials

**Definition 1.** A monic polynomial with the lowest degree in the set of all characteristic polynomial of **a** is said to be the minimal polynomial (MP) of **a** over $F$.

***Note***. The minimal polynomial of the sequence is always a factor of its characteristic polynomials (foundation of parity check attack!).

For Example 5, since $x^3+1 = (x+1)(x^2 + x + 1)$ and $x+1$ does not generate **a**, then $x^2 + x+1$ is the MP of **a**.

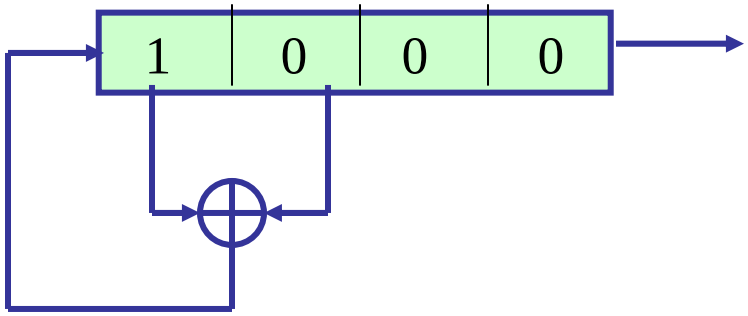
## B. Periods

**Definition 2.** Let $f(x)$ be a polynomial over $F$. A period of $f(x)$ is defined as the smallest positive integer $r$ such that $f(x) \mid x^r - 1$.

**Remark.** Periods of output sequences of an LFSR are completely determined by the characteristic polynomial of the LFSR.

**Property 1.** Let $f(x)$ be the characteristic polynomial of an LFSR, then any output sequence of the LFSR is periodic if and only if $f(0) \neq 0$, i.e., the constant term $c_0$ of $f(x)$ is nonzero.

**Example 6.** For the LFSR in Example 4 in Sec 2.1,




**a** = 00011011011 …

Corresponding characteristic polynomial $f(x) = x^4 + x^3 + x^2$

Since $f(0) = 0$, then **a** is not periodic.

## C. Irreducible Case

**Definition 3.** (Shift equivalent) Let $\underline{\mathbf{a}} = a_0, a_1, \ldots$ and $\underline{\mathbf{b}} = b_0, b_1, \ldots$. If there exists $k \geq 0$, such that

$$b_i = a_{i+k},\ i = 0, 1, \ldots .$$

Then we say that $\underline{\mathbf{a}}$ and $\underline{\mathbf{b}}$ are shift equivalent denoted as $\underline{\mathbf{a}} \sim \underline{\mathbf{b}}$. Otherwise, they are shift distinct.

$S = \{\underline{\mathbf{b}} \mid \underline{\mathbf{b}} \sim \underline{\mathbf{a}}\}$, the set consists of all sequences which are shift equivalent with $\underline{\mathbf{a}}$, is called a shift equivalent class of $\underline{\mathbf{a}}$.

If a sequence is periodic with period $r$, then we may use a vector of dimension $r$ to represent the sequence.

**Example 7.** For $\underline{\mathbf{a}} = (10001)$, $\underline{\mathbf{b}} = (00011)$, and $\underline{\mathbf{c}} = (11110)$, we have

$\underline{\mathbf{a}} \sim \underline{\mathbf{b}}$, but $\underline{\mathbf{a}}$ and $\underline{\mathbf{c}}$ are shift distinct.

**Property 3.** Let $f(x)$ be an irreducible polynomial over $F$ of degree $n$, then the number of shift equivalent classes of non-zero LFSR sequences generated by $f(x)$ is given by

$$(q^n - 1)/\mathrm{per}(f)$$

where per($f$) represents the period of $f(x)$.

**Example 8.** Let $q = 2$ and $f(x) = x^4 + x^3 + x^2 + x + 1$. Then $f(x)$ is irreducible over $GF(2)$. Determine the cycle structure of the LFSR with $f(x)$ as a characteristic polynomial.

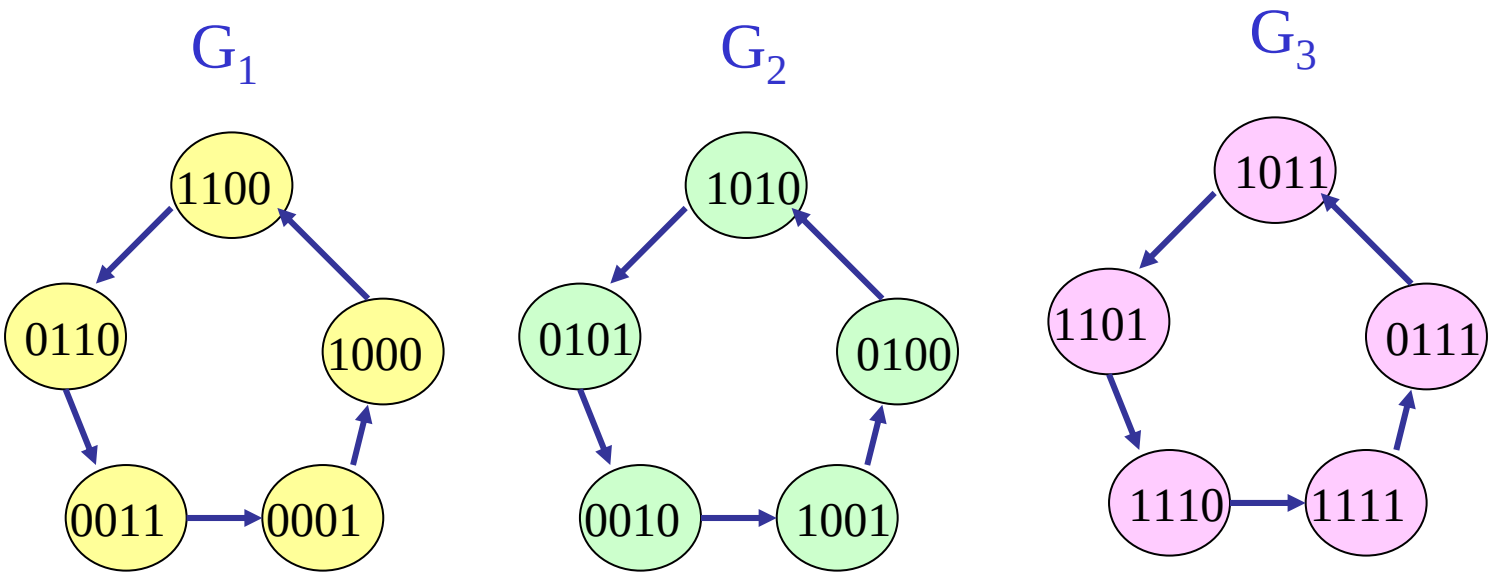
Note that $x^5 + 1 = (x+1)(x^4 + x^3 + x^2 + x + 1)$. Then $f(x) | x^5 + 1 \Rightarrow \text{per}(f) = 5$.

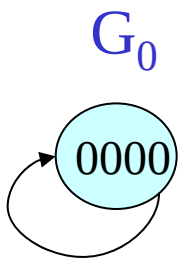
Thus the set consisting of all LFSR sequences generated by $f(x)$ has

$$15/5 = 3$$

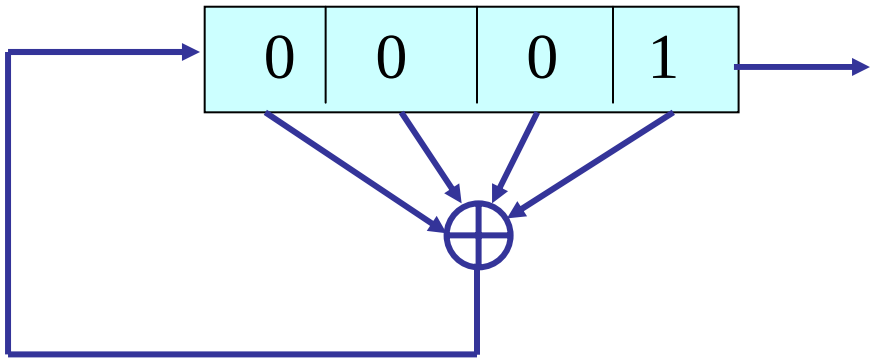
shift equivalent classes for nonzero sequences, which are listed below.

| $G_0$ | $G_1$ | $G_2$ | $G_3$ |
|---|---|---|---|
| 00000 | 00011 | 10010 | 11110 |
| | 00110 | 00101 | 11101 |
| | 01100 | 01010 | 11011 |
| | 11000 | 10100 | 10111 |
| | 10001 | 01001 | 01111 |




State diagram




LFSR with $f(x) = x^4 + x^3 + x^2 + x + 1$

**Definition 4.** A sequence generated by an LFSR over $F = GF(q)$ with period $q^n$-1 is called a *maximal length sequence or an m-sequence* for short.

Let $f(x)$ be irreducible over $F$ with degree $n$. If the period of $f(x)$ is $q^n - 1$, then $f(x)$ is said to be *primitive*.

Thus, if $f(x)$ is primitive, then any non-zero sequence generated by $f(x)$ is an $m$-sequence. So, to generate an $m$-sequence is to find a primitive polynomial!

**Property 4**. If $f(x)$ is primitive, then any nonzero LFSR sequence generated by $f(x)$ has period $q^n$ - 1 and all of them are shift equivalent, *i.e.*, the state diagram has two cycles where one contains zero state and the other contains all non states.

**Example 9.** Let $q = 2$.

- $n = 3$, for the LFSR in Example 2, we have

$$f(x) = x^3 + x + 1$$

then $f(x)$ is primitive, i.e., the period of $f(x)$ is equal to 7.

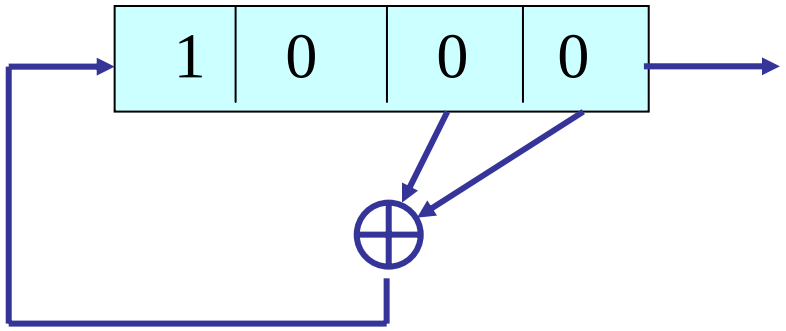
In the following, we use $L$ to represent the shift operator. Then

$$\begin{aligned}
\underline{\mathbf{a}} &= 1001011 \\
L\underline{\mathbf{a}} &= 0010111 \\
L^2\underline{\mathbf{a}} &= 0101110 \\
L^3\underline{\mathbf{a}} &= 1011100 \\
L^4\underline{\mathbf{a}} &= 0111001 \\
L^5\underline{\mathbf{a}} &= 1110010 \\
L^6\underline{\mathbf{a}} &= 1100101
\end{aligned}$$

All nonzero states of the LFSR are in one cycle.

- $n = 4$, $f(x) = x^4 + x + 1$, primitive over $GF(2)$.

  00010 01101 01111, period 15.

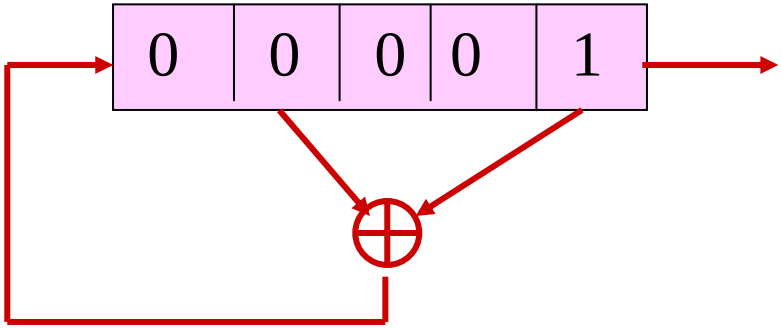



- $n = 5$, $f(x) = x^5 + x^3 + 1$, primitive over $GF(2)$.

  1 0 0 0 0 1 0 1 0 1

  1 1 0 1 1 0 0 0 1 1

  1 1 1 0 0 1 1 0 1 0 0, period 31.



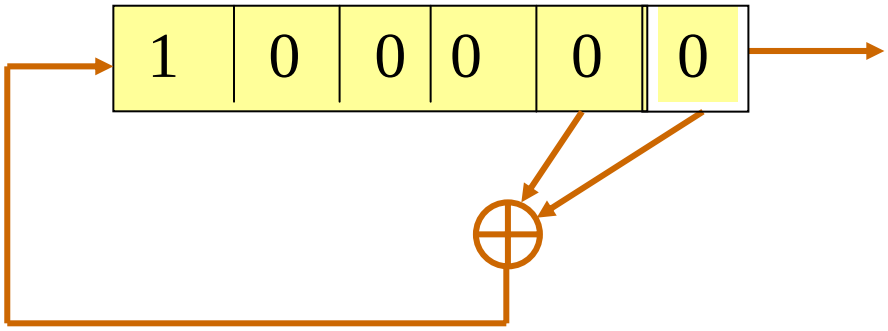

- $f(x) = x^6 + x + 1$, primitive over $GF(2)$.

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

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

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

  period 63.



# 3. Randomness Measurements

## A. Definitions of some basic concepts

Let **a** = $(a_0, a_1, \ldots, a_{N-1})$ be a binary sequence of period $N$.

**Run:** For a binary sequence **a** with period $N$, $k$ consecutive zeros (ones) is called a run of zeros (or ones) of length $k$.

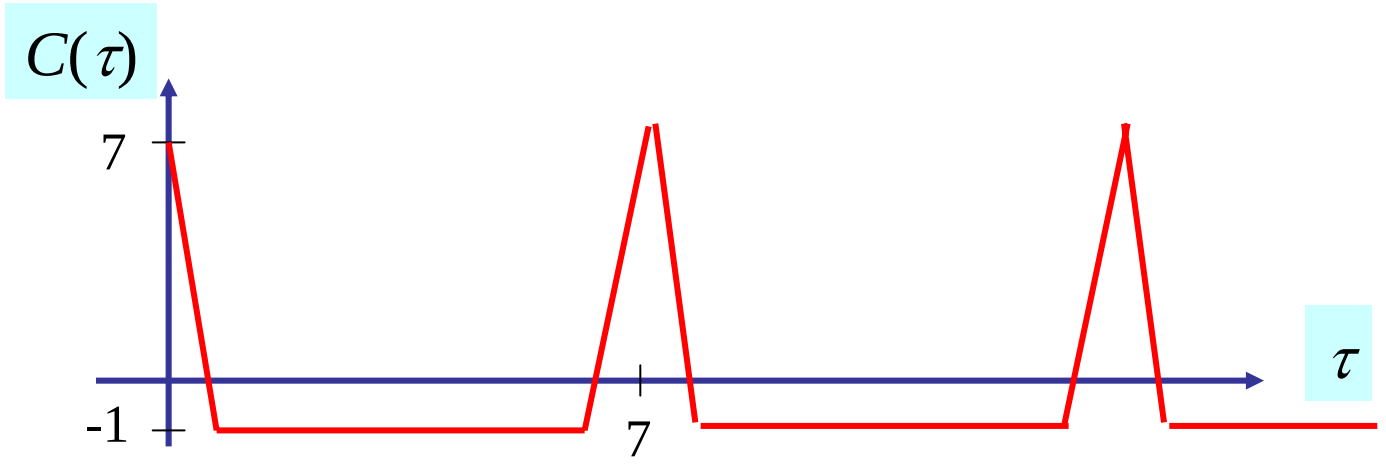
**Autocorrelation:** The auto correlation function $C(\tau)$ of **a** is defined as

$$C(\tau) = \sum_{i=0}^{N-1} (-1)^{a_i + a_{i+\tau}}, \tau = 0,1,\cdots.$$

which measures the difference between agreements and disagreements between the sequence and its shifted version.

**Example 10**. Let **a** = (1001011) with period 7. Then

$$C(\tau) = \begin{cases} 7 & \text{for } \tau \equiv 0 \bmod 7 \\ -1 & \text{for } \tau \neq 0 \bmod 7 \end{cases}$$



# Cross correlation between two sequences:

Let $\underline{\mathbf{b}} = (b_0, b_1, \ldots, b_{N-1})$ be another binary sequence of period $N$. The cross correlation of $\underline{\mathbf{a}}$ and $\underline{\mathbf{b}}$ is defined by

$$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(\tau) = \sum_{i=0}^{N-1} (-1)^{a_i + b_{i+\tau}}$$

**Example 11**. With $\underline{\mathbf{a}}$ in Example 10, let $\underline{\mathbf{b}} = (1110100)$. Then

$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(0) = 1\times(-1)^0 + 6\times(-1) = -5$

$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(1) = 5\times(-1)^0 + 2\times(-1) = 3$

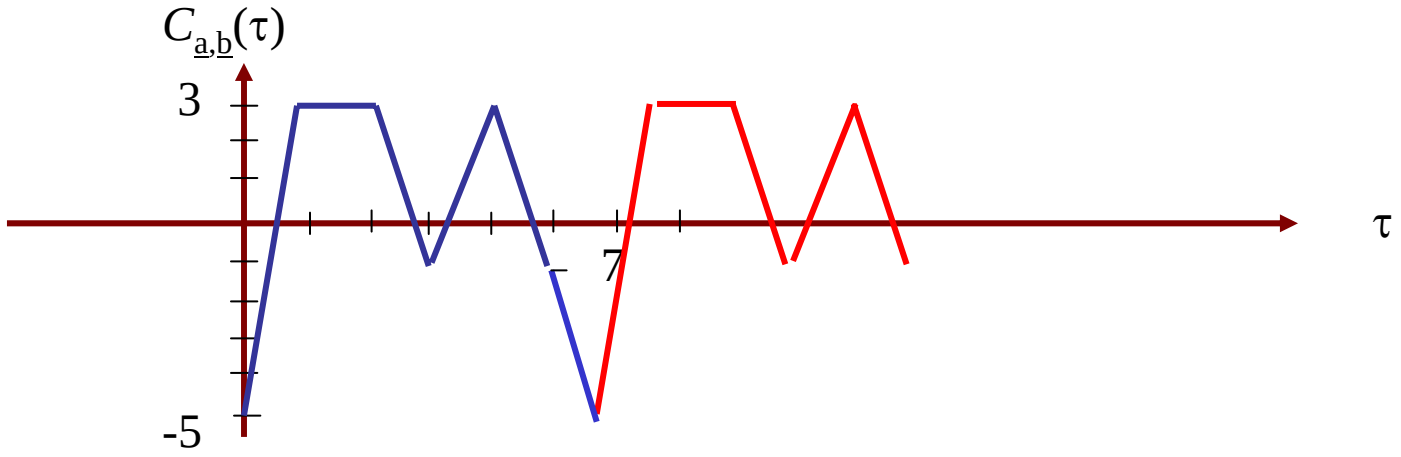
$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(2) = 3$

$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(3) = -1$

$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(4) = 3$

$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(5) = -1$

$C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(6) = -1$



# B. Golomb's Three Randomness Postulates

**R1.** Balance property: the number of 0's is nearly equal to the number of 1's. Precisely, $\left|\sum_{i=0}^{N-1}(-1)^{a_i}\right| \le 1$

**R2.** Run property: In each period, half runs have length 1

1/4 runs have length 2

1/8 runs have length 3

...

Moreover, for each these lengths, there are equally many runs of 0's and that of 1's.

**R3.** 2-level autocorrelation:

$$C(\tau) = \begin{cases} N & \text{for } \tau \equiv 0 \bmod N \\ -1 & \text{for } \tau \neq 0 \bmod N \end{cases}$$

# C. (Unconditional) Randomness Criteria:

(A) Long period

(B) Statistic Properties:

(1) Balanced property: each element occurs nearly equally many times.

(2) Run property R2.

(3) $k$-tuple distribution: for $k = \log N$, each $k$-tuple $d_0, d_1, ..., d_{k-1}$, $d_i \in GF(q)$, occurs nearly equally many times in one period.

(C) Correlation

(1) 2-level auto correlation

(2) Low cross correlation: $0 \le |C_{\underline{\mathbf{a}},\underline{\mathbf{b}}}(\tau)| \le \sigma\sqrt{N}$, where $\sigma > 0$ is a constant.

(D) Linear span : The degree of the minimal polynomial of $\underline{\mathbf{a}}$ is said to be a linear span of $\underline{\mathbf{a}}$. Thus the linear span of $\underline{\mathbf{a}}$ is the shortest length of LFSR that generates $\underline{\mathbf{a}}$, denoted as $LS(\underline{\mathbf{a}})$. This can be computed by the Berlekamp-Massey algorithm (see Lecture Notes). Large Linear span:

$$Nt/2 \le LS(\underline{\mathbf{a}}) \le N \quad \text{where } 0 < t < 1, \text{ constant.} \qquad \text{(B)}$$

Let $\rho = LS(\underline{a})/N$ (normalized linear span), then (B) states that $t/2 < \rho < 1$.

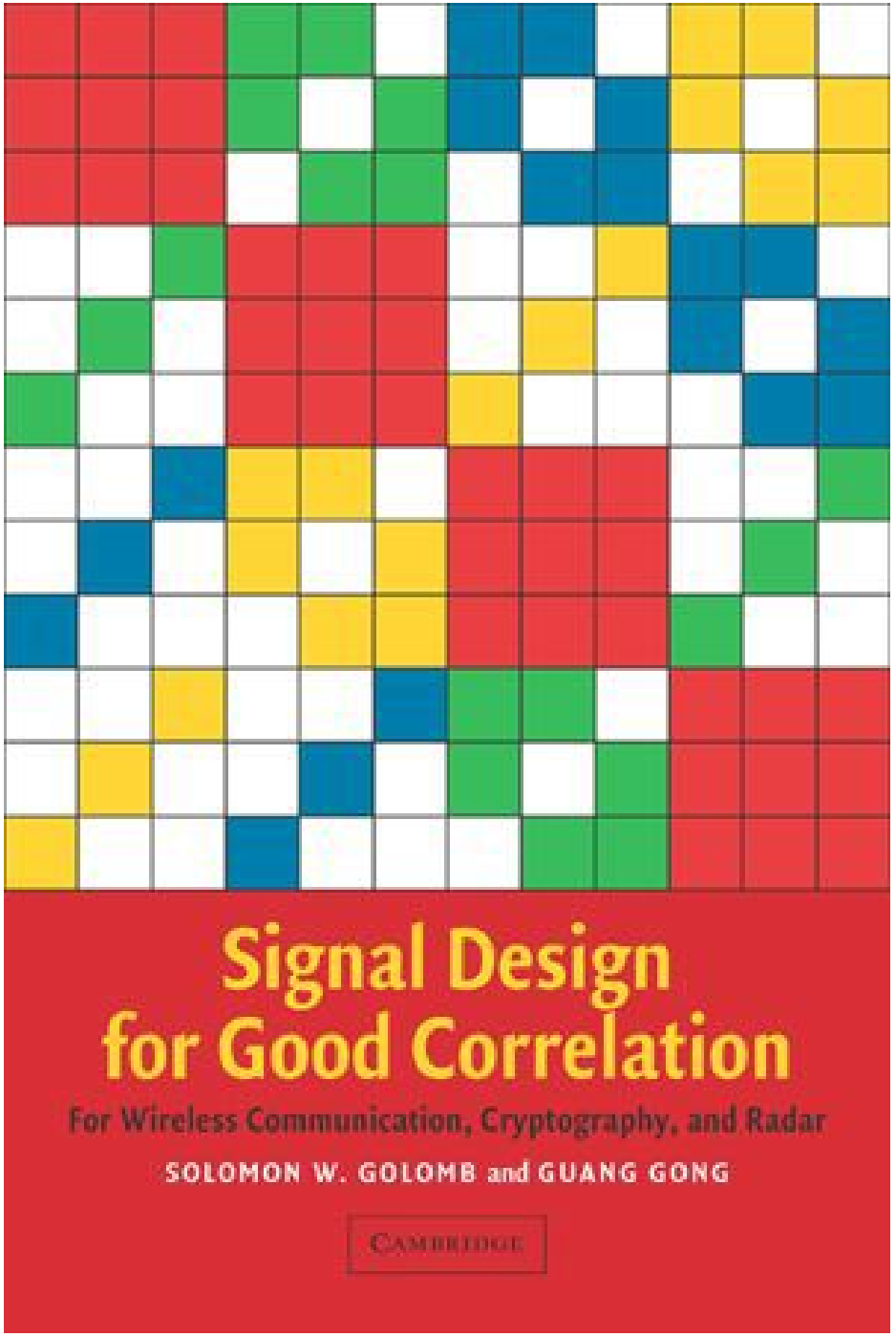
# D. Profiles of Binary M-sequences of Degree *n*

| | |
|---|---|
| Period | $2^n - 1$ |
| Balance | $2^{n-1}$ 1's and $2^{n-1}$ -1 0's |
| Run property | (1) For $1 \le k \le n-2$ runs of 0's (1's) of length $k$ occurs $2^{n-2-k}$ times<br>(2) Zero run of length $n$-1 occurs once; no runs of 1's of length $n$-1<br>(3) Runs of 1's of length n occurs once |
| Span *n* property or ideal *n*-tuple distribution | Each nonzero *n*-tuple occurs once |
| Autocorrelation | 2-level |
| Linear span and $\rho$ | *n*, the normalized linear span $\rho = \frac{n}{2^n - 1}$ |

As good as it could be!

Only need to know 2*n* bits to reconstruct the entire sequence!

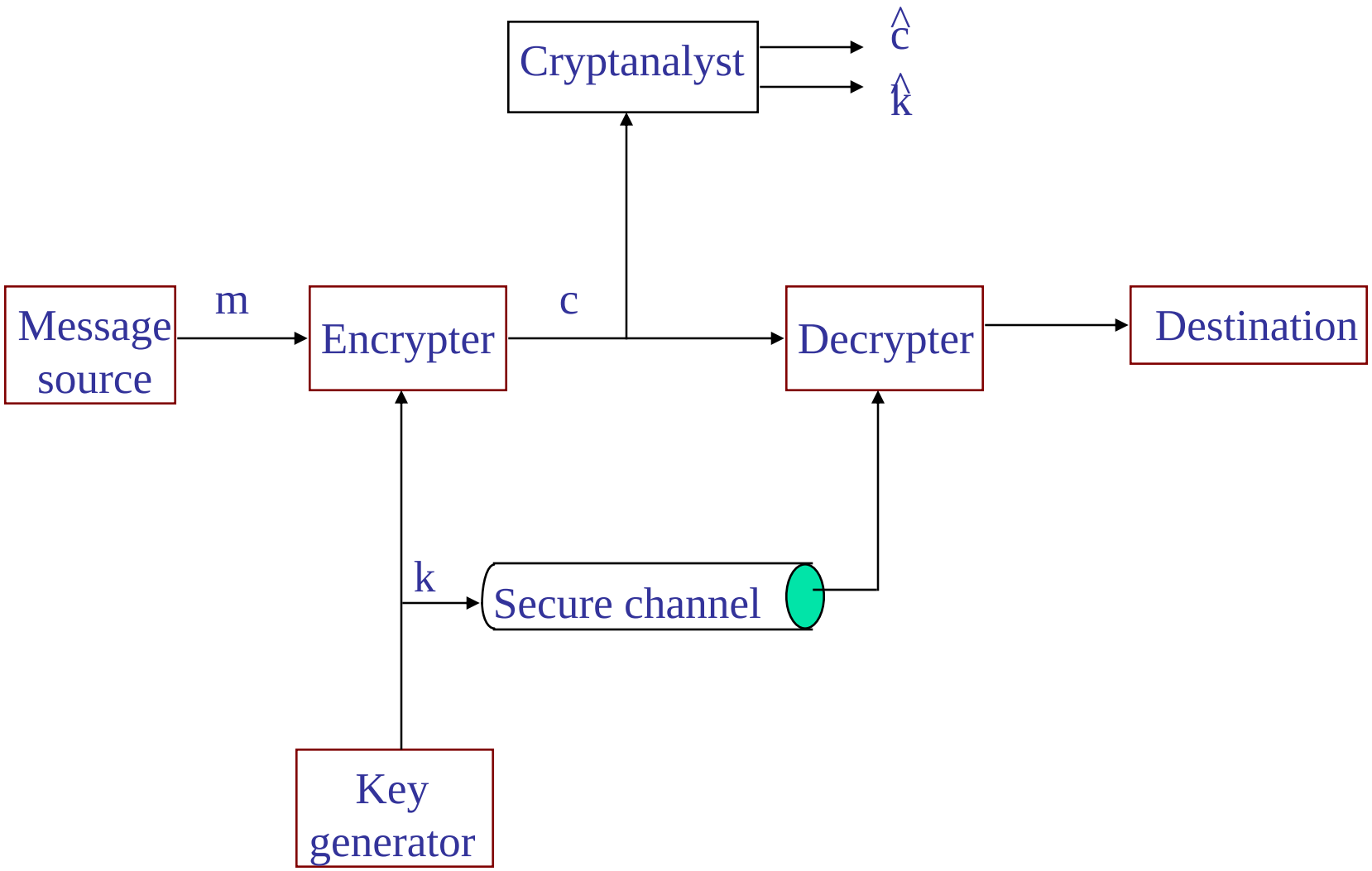
**For more about pseudo-random sequences, see Golomb and Gong's new book:**



University of Waterloo

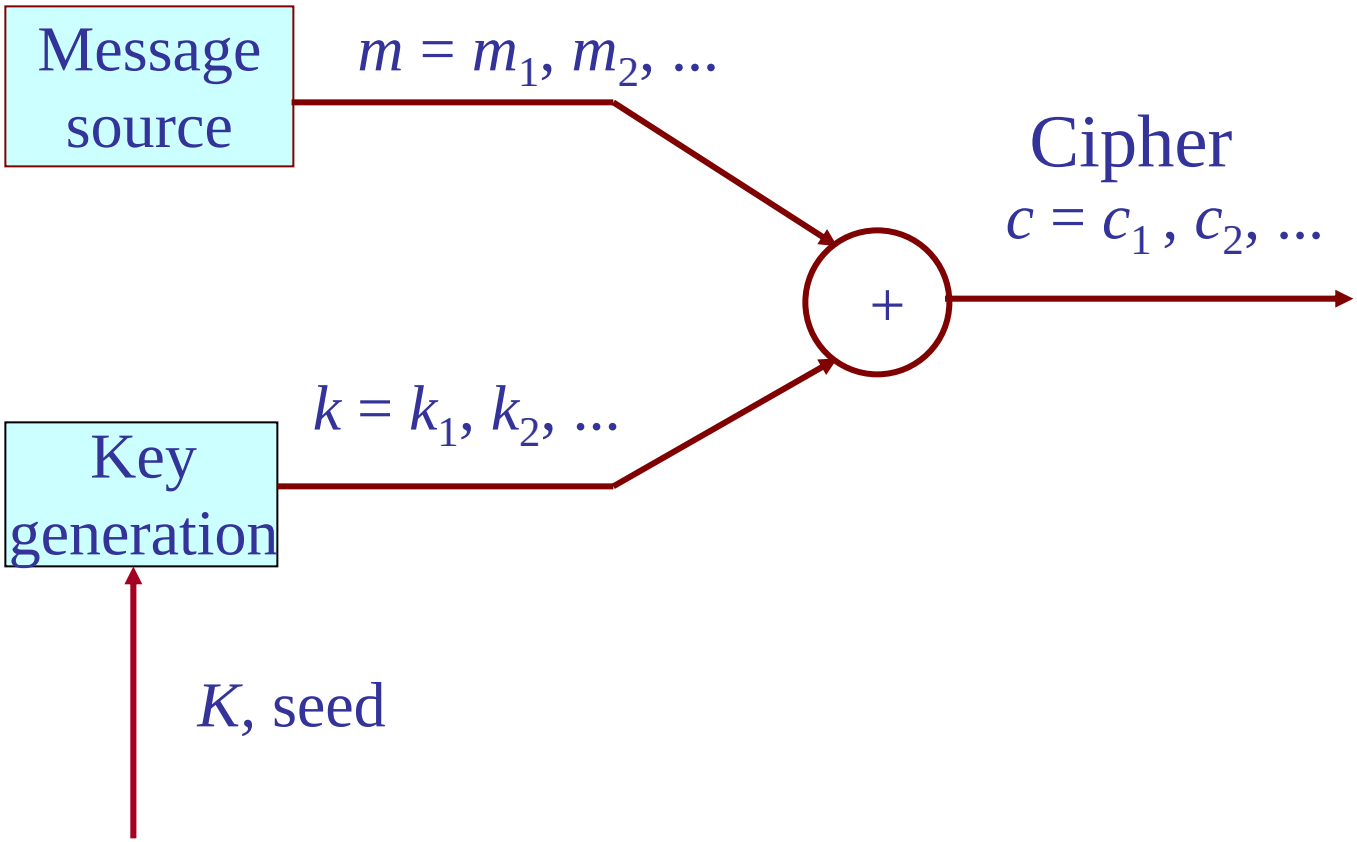
# 4. Design of Pseudorandom Sequence Generators (PRSGs) Towards Large Linear Span

- One-time-pad and Design Principles of Stream Ciphers
- Known Approaches for Design of PRSG Based on LFSR




**Model of Conventional Cryptosystem**

# Model of Stream Cipher



# One-time-pad and Design Principles of Stream Ciphers

- One-time-pad
- Randomness Measurements for PRSG
- Known Approached for Key Generators for Stream Ciphers

# One-time-pad

One-time-pad means that different messages are encrypted by different key streams.

- Shannon's Result (1948): One-time-pad is unbreakable.
  Request for large period.

Massey's Discovery (1969):
If a binary sequence has linear span $n$ then the entire sequence can be reconstructed from $2n$ consecutive known bits by the Berlekamp-Massey algorithm.
Request for large linear span.

Berlekamp-Massey algorithm attack (1969)

LFSR generates $m$-sequences

= Maximal length sequences

= Pseudo-noise (PN) sequences

= PSG (1948-1969)

# Randomness Measurements

## Randomness Measurements for PSG:

### Mathematical Properties:

- Long Period
- Balance Property
- Run Property
- *n*-tuple Distribution
- Two-level Auto  Correlation and Low Cross Correlation
- Large Linear Span

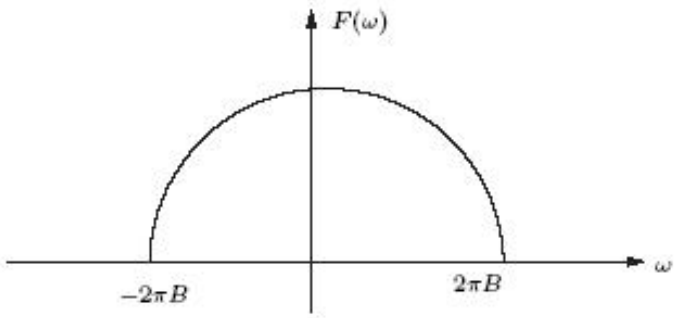
### Computational Complexity Property:

- Indistiguinishability:  a sequence, schematically  generated by an algorithm or a device,  cannot be distiguinished  from a truly random sequence in terms of any polynomial algorithm with negligible probability.

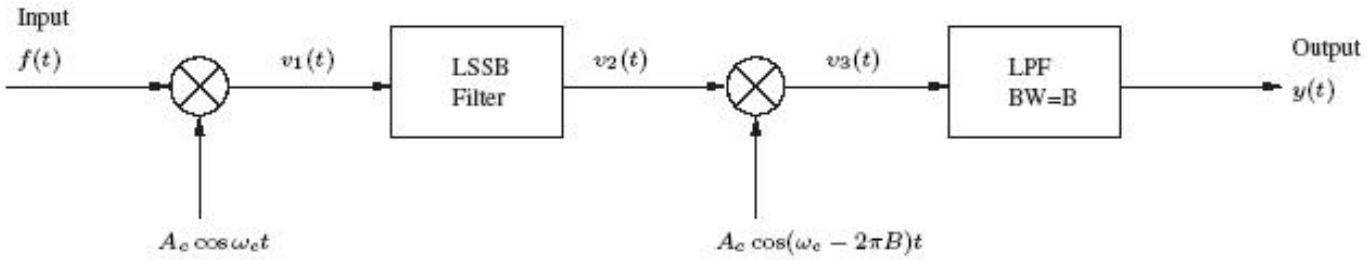
# Known Approached for Design of Key Generators in Stream Ciphers

- Analog Voice Encryption: Shuffling Frequency Spectra (40's-50)
- LFSR Based Generators:
  - Linear Feedback Shift Registers (LFSR) (1948-1969)
  - Filter Function Generators (Key: 1973)
  - Combinatorial Function Generators (Groth: 1971)
  - Clock Controlled Generators (Beth and Piper: 1984
  - Shrinking Generators (Coppersmith-Krawczys-Mansour, 93)
- Block Ciphers Used as Stream Cipher Modes
  - Cipher Feedback Mode (CFB)
  - Counter Mode (CTR)

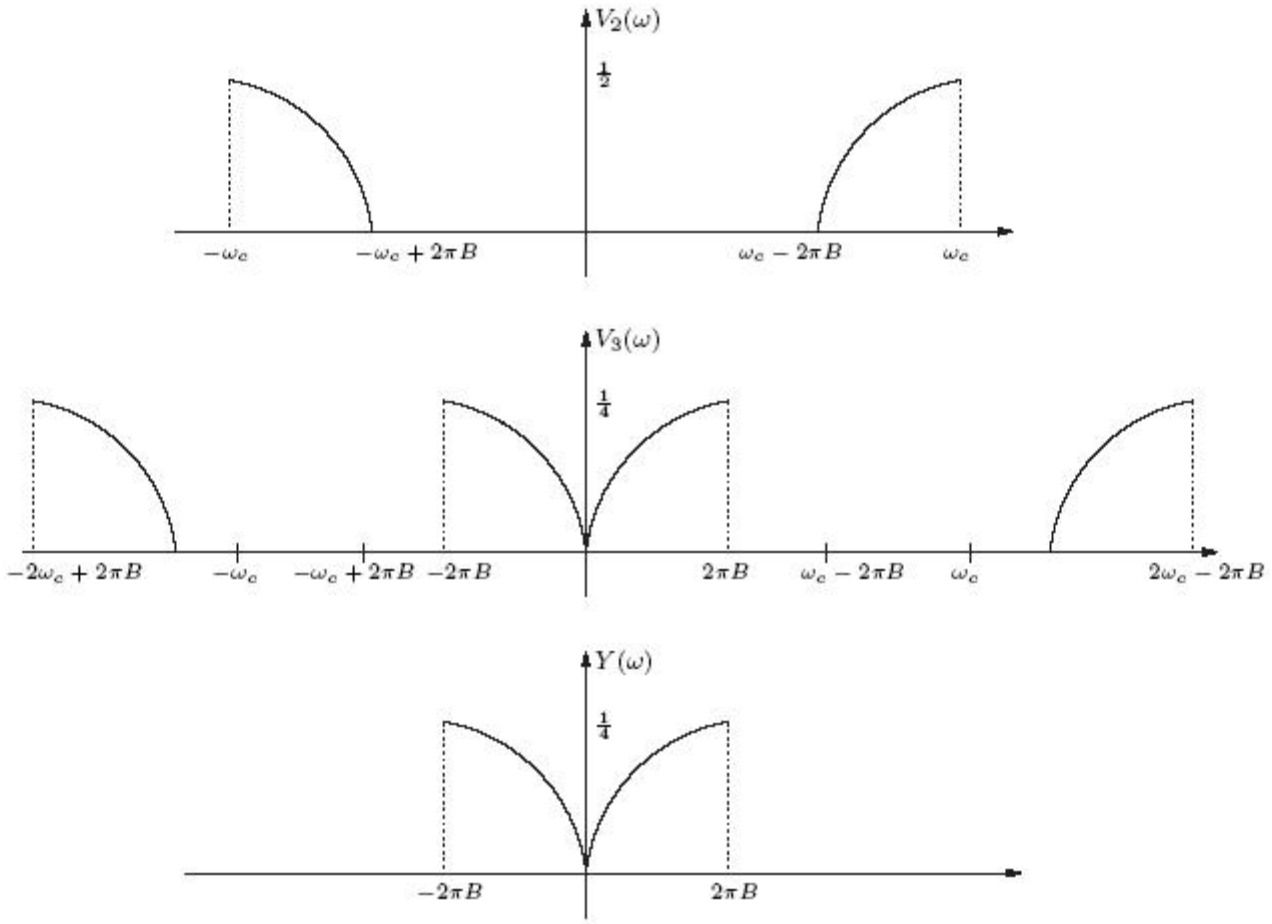
# An Example of Analog Voice Encryption



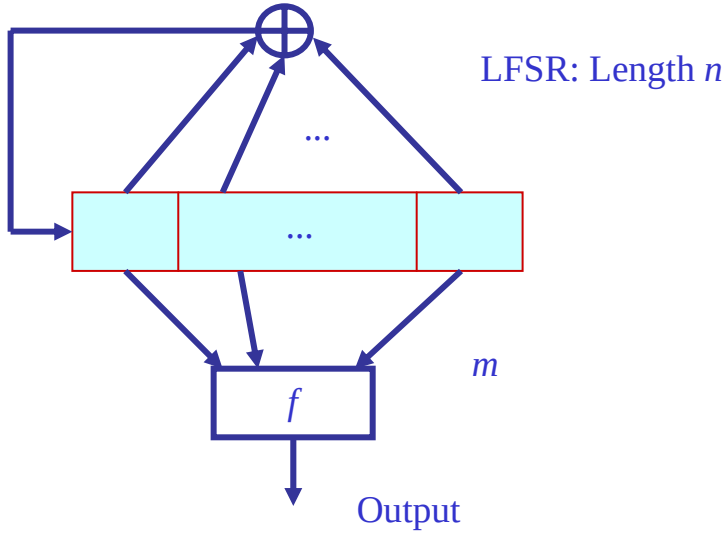

(a)



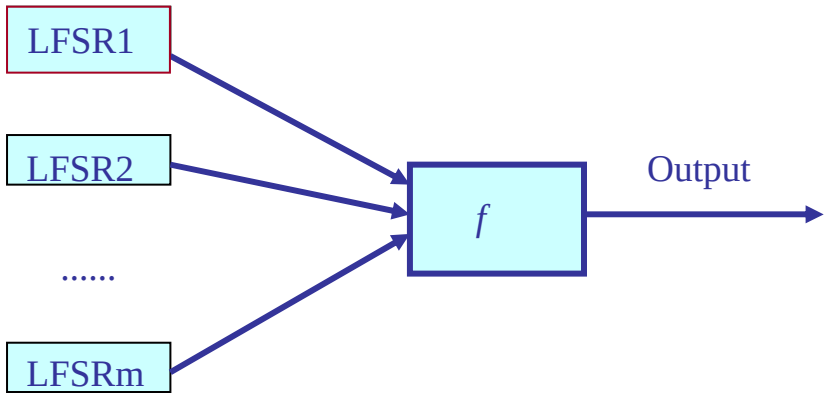

(b)



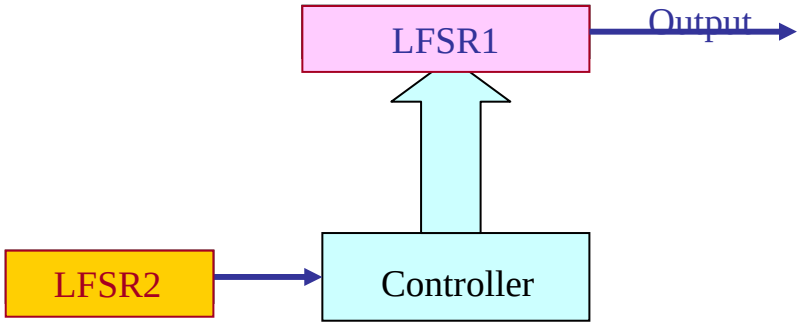

# Frequency Spectra



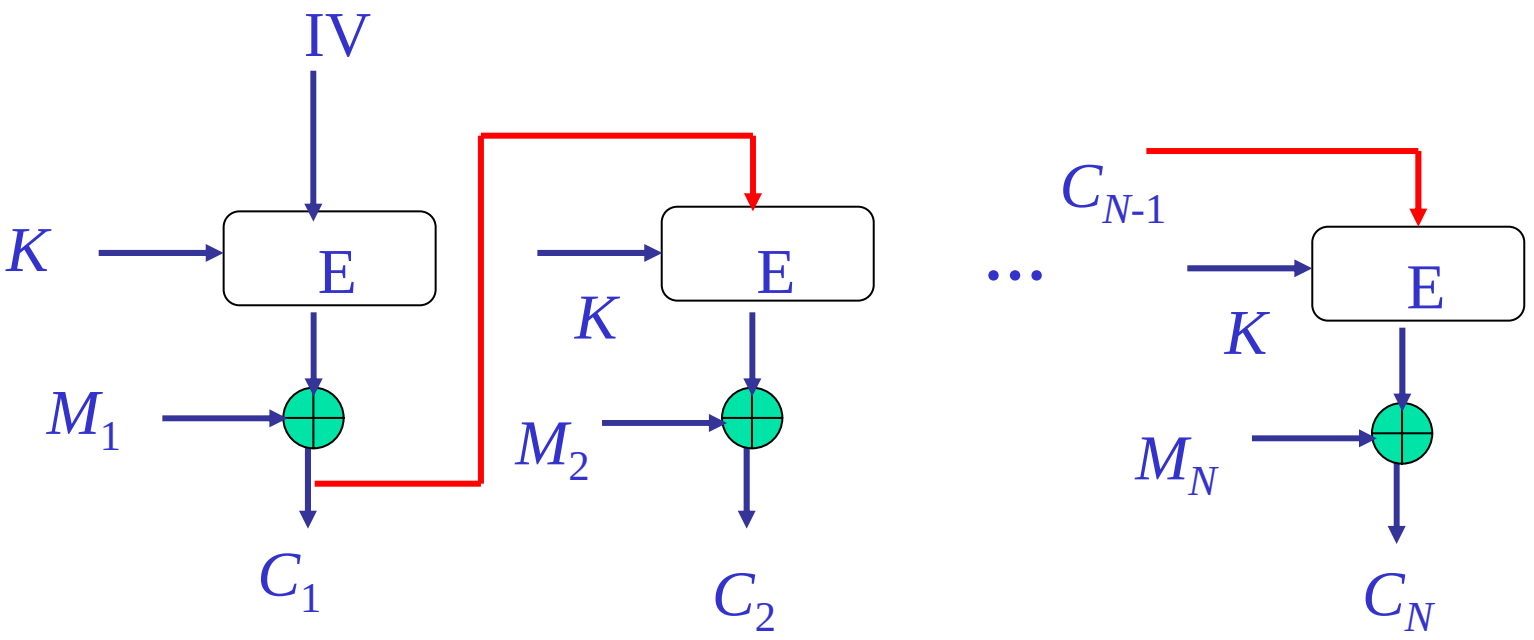

**A Filtering Generator**




**A Combinatorial Function Generator**



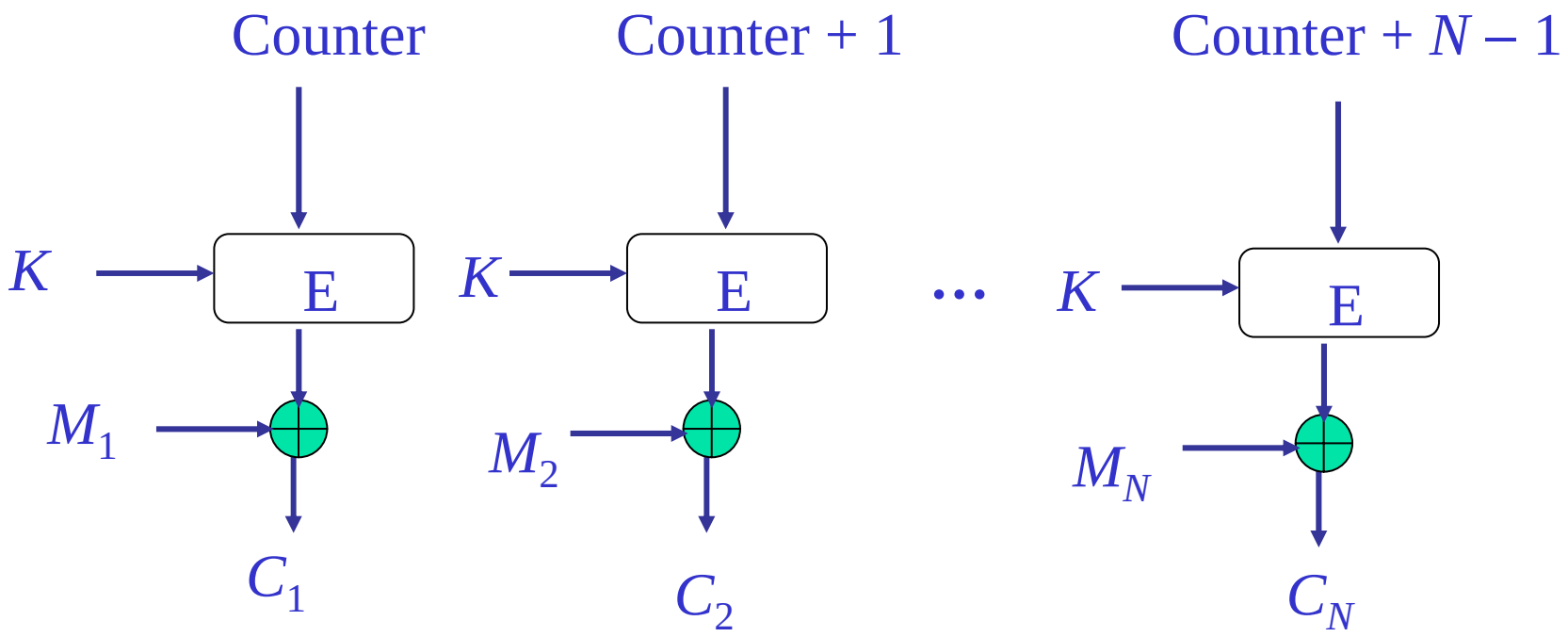

**Clock-control /Shrinking Generators**




$$K_1 = E_K(\text{IV}), \text{ and } K_i = E_K(C_{i-1}), \quad i = 2, 3, ..., N$$

$$C_i = K_i + M_i, \quad i = 1, 3, ..., N$$

**Block Cipher E Used as CFB Mode (Stream Cipher Mode)**




$$K_i = E_K(\text{Counter} + i - 1), \quad i = 1,\ 2, ..., N$$

$$C_i = K_i + M_i, \quad i = 1,\ 3, ..., N$$

**Block Cipher E Used as Counter Mode (Stream Cipher Mode)**

# 5. Examples of Stream Ciphers in Practice

- Attacks and Outline of Stream Ciphers in Practice
- A5/1 in GSM System
- W7: analogue of A5/1
- WG: A Stream Cipher Candidate from ECRYPT

# Attacks on Stream Ciphers

- ✓ Correlation Attacks
- ✓ Time/Memory/Data Tradeoff Attacks:
- ✓ Guess and Determine Attacks
- ✓ Distinguish Attacks
- ✓ Algebraic Attacks

These attacks lead to investigation of boolean functions with the properties against the above attacks.

# Stream Ciphers in Practice

Stream Ciphers in Various Applications:

- A5/1 in GSM System (secret key can be recovered in a few seconds on a PC)
- W7: analogue of A5/1
- RC4 in Web Security (some distinguish attacks)
- E0 in Blue Tooth Standard (using algebraic attack to get key needs computation of $2^{49}$ & data 20kb)

Stream Ciphers in W-CDMA and CDMA 2000:

- LFSR based Stream Ciphers in CDMA 2000
- Block Ciphers Used in Stream Cipher Modes

  - Kasumi in W-CDMA (or UMTS) (CFB Mode)
  - AES RIJDAEL in CDMA2000 (also in WEP)

    (CRT Mode)

# Encryption Algorithms in CDMA2000

CDMA2000 systems proposed to employ three different encryption algorithms:

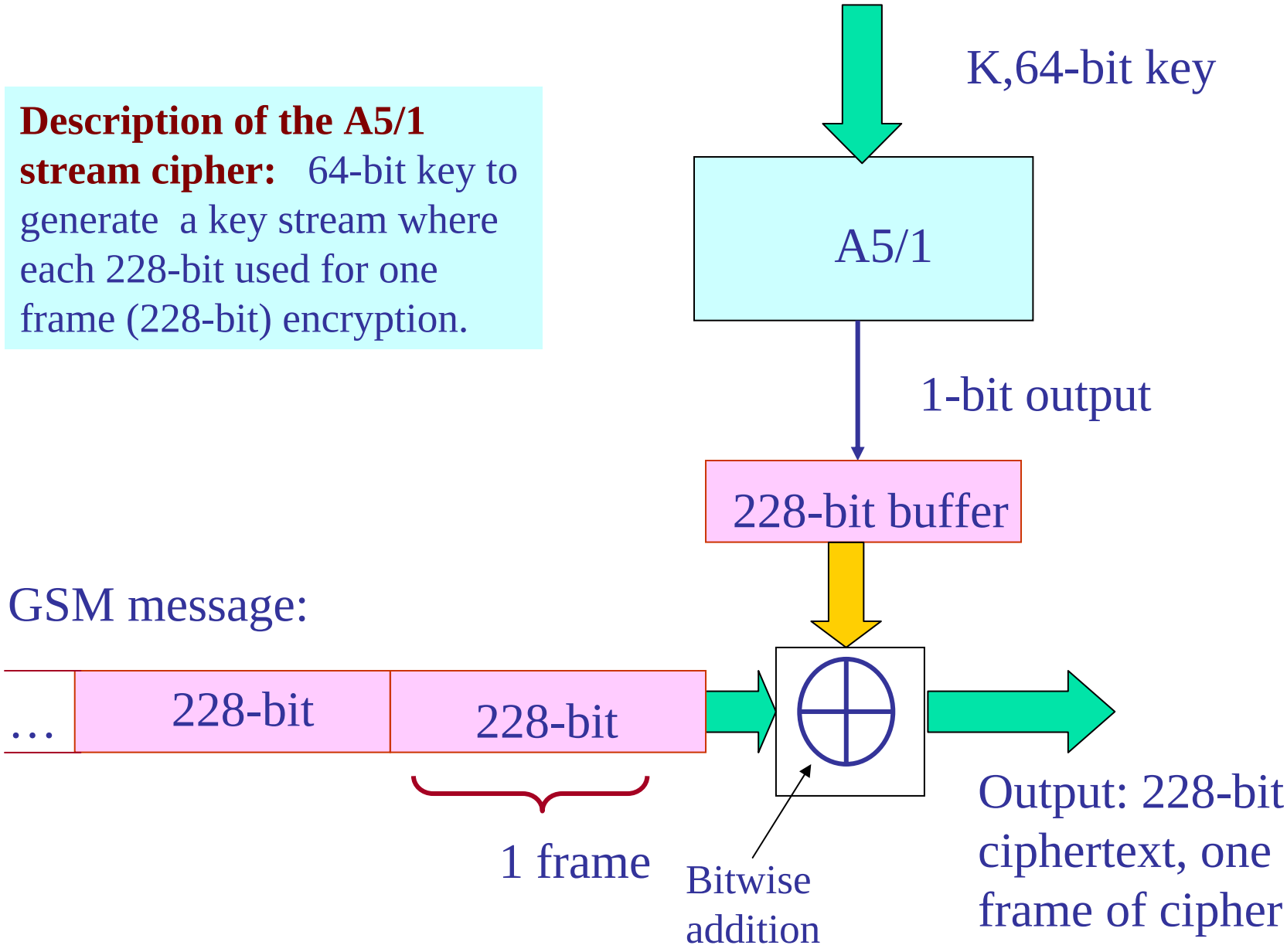
- Cellular Authentication and Voice Encryption (CAVE)
- Cellular Message Encryption Algorithm (CMEA - for signaling)
- ORYX (for data) on the CDMA channel.

All of these three ciphers are LFSRs based stream ciphers and all of them suffered serious attacks due to none of them are discussed in publicity.

## A. A5/1 stream cipher key generator for secure GSM conversations

**Note.** A GSM conversation is sent as a sequence of frames per 4.6 millisecond, and each frame contains 228 bits.

**Description of the A5/1 stream cipher:** 64-bit key to generate a key stream where each 228-bit used for one frame (228-bit) encryption.



# Construction of A5/1 Generator:

## Parameters:

(a) Three LSFRs which generate *m*-sequences with periods $2^{19} - 1$, $2^{22} - 1$, $2^{23} - 1$, respectively.

1. LFSR 1: $f_1(x) = x^{19} + x^5 + x^2 + x + 1$ generates **a** = $\{a(t)\}$.

2. LFSR 2: $f_2(x) = x^{22} + x + 1$ generates **b** = $\{b(t)\}$.

3. LFSR 3: $f_3(x) = x^{23} + x^{16} + x^2 + x + 1$ generates **c** = $\{c(t)\}$.

4. Tap positions: $d_1 = 11$, $d_2 = 12$ and $d_3 = 13$.

(b) Majority function $f(x_1, x_2, x_3) = (y_1, y_2, y_3)$ is defined by

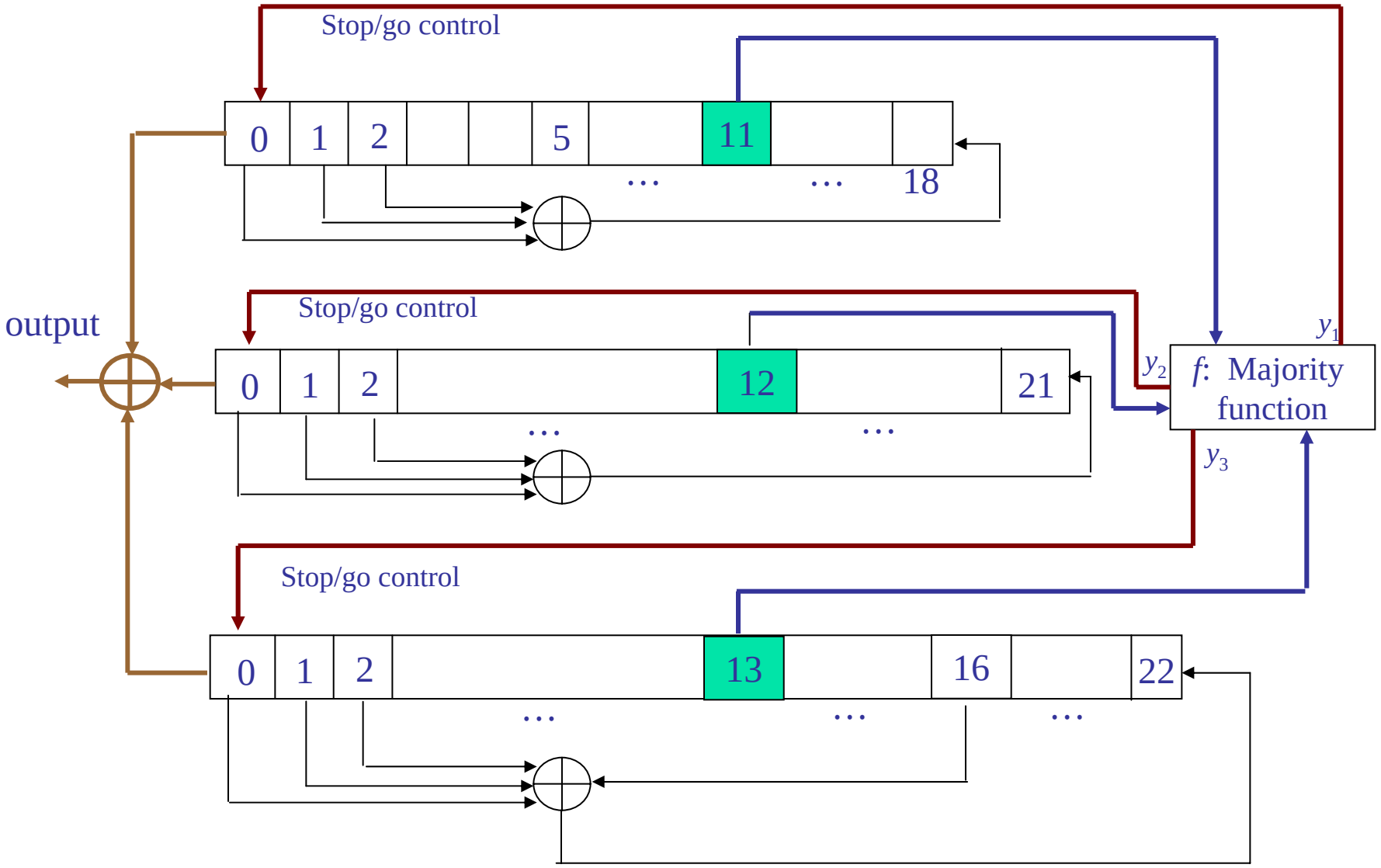
| $f(a(t+11), b(t+12), c(t+13)) = (y_1, y_2, y_3)$ | $a(t+11)$ | $b(t+12)$ | $c(t+13)$ |
|---|---|---|---|
| (1,1,1) | 0<br>1 | 0<br>1 | 0<br>1 |
| (1,1,0) | 0<br>1 | 0<br>1 | 1<br>0 |
| (0,1,1) | 0<br>1 | 1<br>0 | 1<br>0 |
| (1,0,1) | 1<br>0 | 0<br>1 | 1<br>0 |

**Output:**

The output sequence **u** = $\{u(t)\}$ which performs at time $t$,

$$u(t) = a(i_1) + b(i_2) + c(i_3),\ t = 0, 1, \ldots$$

where $i_1$, $i_2$, and $i_3$ are determined in a stop-and-go clock controlled model by the majority function $f$.




A5/1 Key Stream Generator

For example, at time $t$, if

$f(a(t+11), b(t+12), c(t+13)) = (1, 1, 0)$

i.e., $(y_1, y_2, y_3)=(1, 1, 0)$, then LFSR 1 and LFSR 2 are clocked and LFSR 3 has no clock pulse.

**Session key or seed:** initial states for three LFSRs, a total of 64 bits.

**Note 2.** The first 'original' A5 algorithm was renamed A5/1. Other algorithms include A5/0, which means no encryption at all, and A5/2, a weaker over-the-air privacy algorithm. Generally, the A5 algorithms after A5/1 have been named A5/x. Most of the A5/x algorithms are considerably weaker than the A5/1. A5/3 is available in the work group of wireless communications

## What does A5/1 suffer ?

- It can be broken with few hours by a PC.
- Short period problem: Without stop/go operation, the period of sum of the three LFSRs is given by

$$(2^{19}-1)(2^{22}-1)(2^{23}-1).$$

However, the experiement shows that the period of A5/1 is arround

$$(4/3)(2^{23}-1).$$

- Collision problem: different seeds (i.e., different initial states of three LFSRs) may result in the same key stream (our new results shows that only 70% seeds produce different key streams.)
- The maority function is the worst function in terms of correlation with all affine functions.

# Possible Attacks on A5/1

- **Brute-Force Attack against A5**

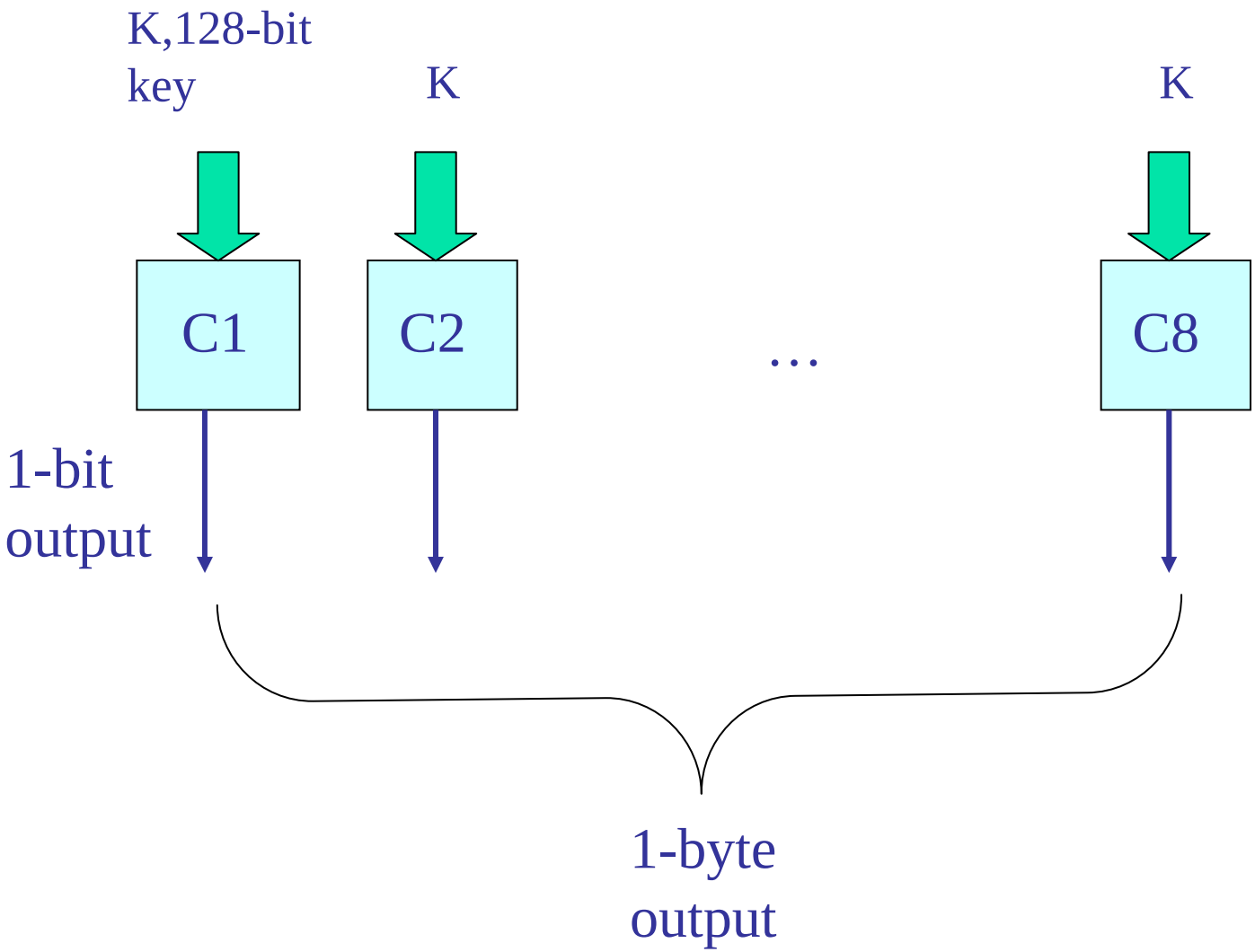
  If we have a Pentium III class chip with approximately 20 million transistors and the implementation of one set of LSFRs (A5/1) would require about 2000 transistors, we would have a set of 10,000 parallel A5/1 implementations on one chip. If the chip was clocked to 600 MHz, we could try approximately 2M keys per second per A5/1 implementation. A key space of $2^{54}$ keys would thus require about 900,000 seconds, 250 hours, with one chip.

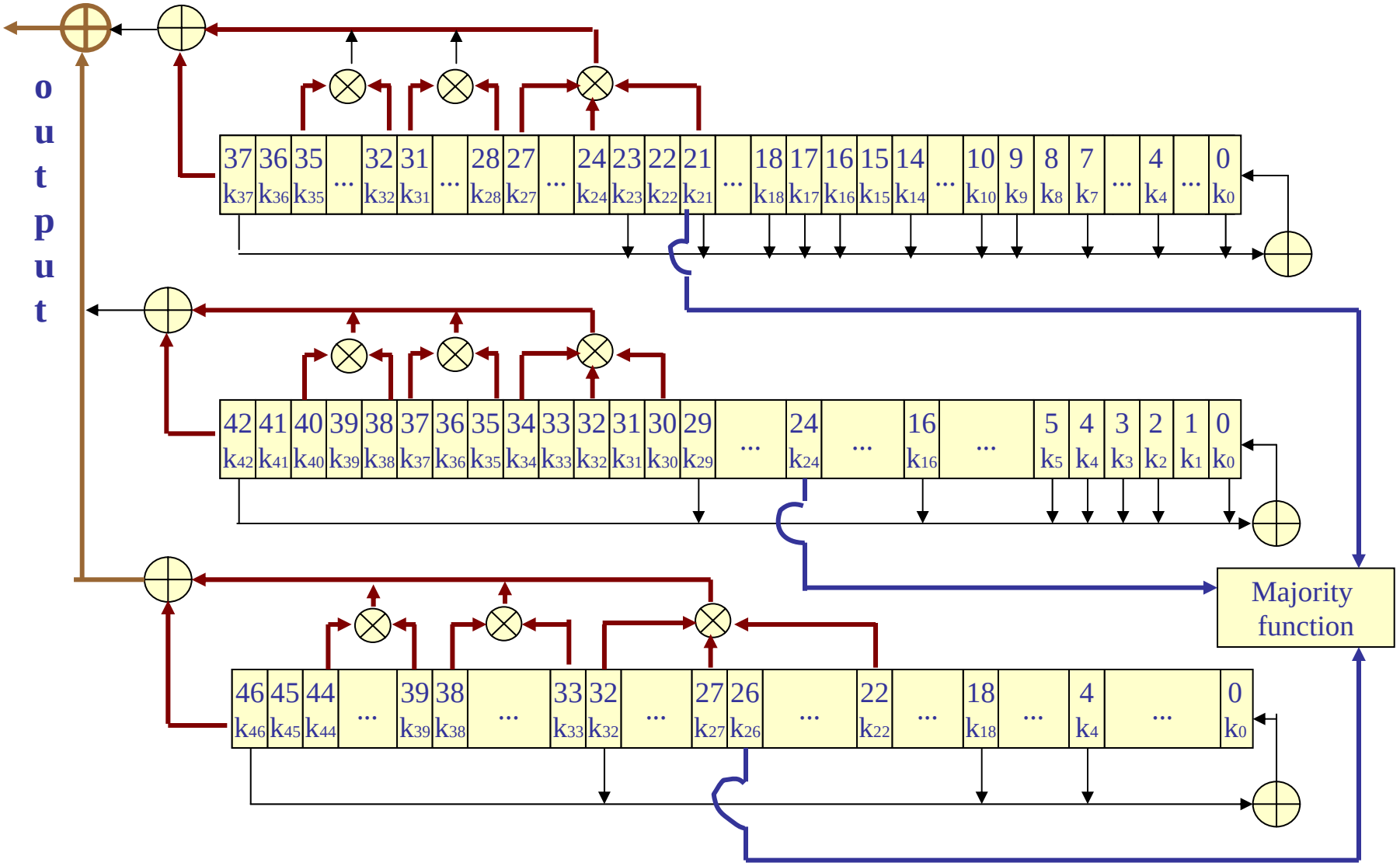
- Alex Biryukov and Adi Shamir (co-inventor of the RSA) claim to be able to penetrate the security of a A5/1 ciphered GSM call in less than one second using a PC with 128 MB RAM and large hard drives.

# B. w7 -- an Analogue Cipher of A5

w7 stream cipher algorithm is proposed by S. Thomas, D. Anthony, T. Berson , and G. Gong published as an INTERNET DRAFT, April 2002.

**Description of w7:** The w7 algorithm is a byte-wide, synchronous stream cipher optimized for efficient hardware implementation at very high data rates. It is a symmetric key algorithm supporting key lengths of 128 bits. It contains eight similar models, C1, C2, …, C8 where C2 is illustrated as follows.


K,128-bit
key
K
K
C1
C2
…
C8
1-bit
output
1-byte
output




# The W7 Cipher Algorithm

# C. WG: A Stream Cipher Candidate from ECRYPT

ECRYPT is a Network of Excellence within the Information Societies Technology (IST) Programme of the European Commission. The ECRYPT Stream Cipher Project is a multi-year (2004-2008) project to identify new stream ciphers that might become suitable for widespread adoption. There are 34 submissions. The final report will be made in January 2008.

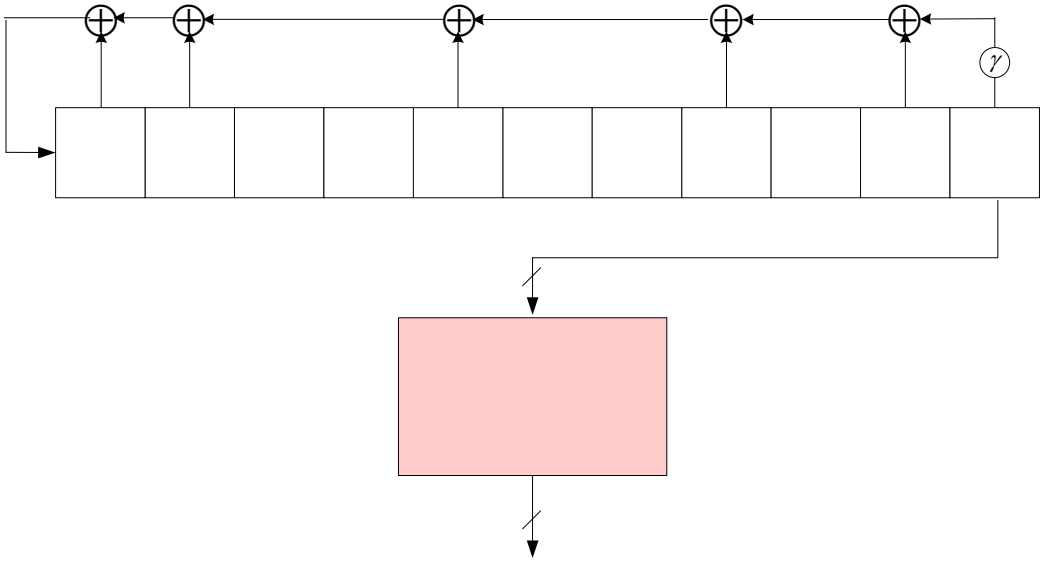
- WG Steam Cipher, Yassir Nawaz and Guang Gong, 2005.

University of Waterloo

# WG Cipher Outline

- Synchronous stream cipher submitted in Profile 2 (for hardware applications)
- Key lengths of 80, 96, 112 and 128 bits
- IVs of 32, 64 bits and also the same lengths as the key are allowed
- Estimated strength $2^{128}$ (exhaustive search)
- Based on Welch Gong (WG) transformation sequences, well studied in sequence design for communications
- The keystream has the cryptographic properties of WG sequences

# WG Cipher Block Diagram




- LFSR generates an *m*-sequence over GF($2^{29}$)
- M-sequence is filtered by a WG transformation, GF($2^{29}$) $\rightarrow$ GF(2), to produce a running keystream

# WG Transformation

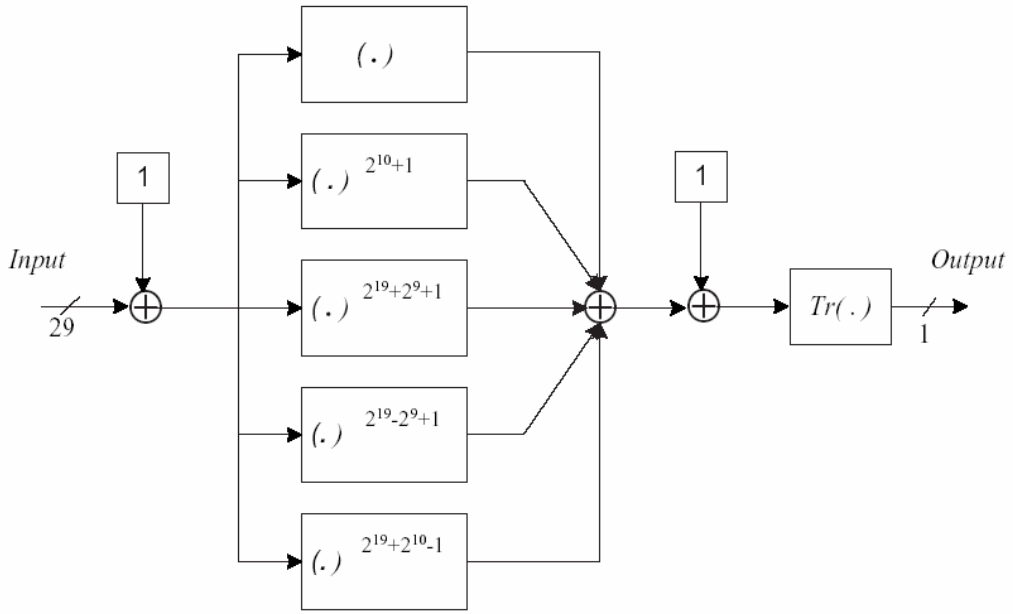
Mathematical description of the WG transformation $f(x)$: It is a function from GF($2^{29}$) to GF(2) given by

$$f(x) = Tr(t(x+1)+1) \quad \text{where}$$

$$t(x) = x + x^{q_1} + x^{q_2} + x^{q_3} + x^{q_4} \quad \text{where}$$

$$q_1 = 2^{19} + 2^9 + 1, \qquad q_2 = 2^{19} - 2^9 + 1$$
$$q_3 = 2^{19} + 2^{10} - 1, \qquad q_4 = 2^{10} + 1$$

and tr($x$) is the trace function from GF($2^{29}$) to GF(2).

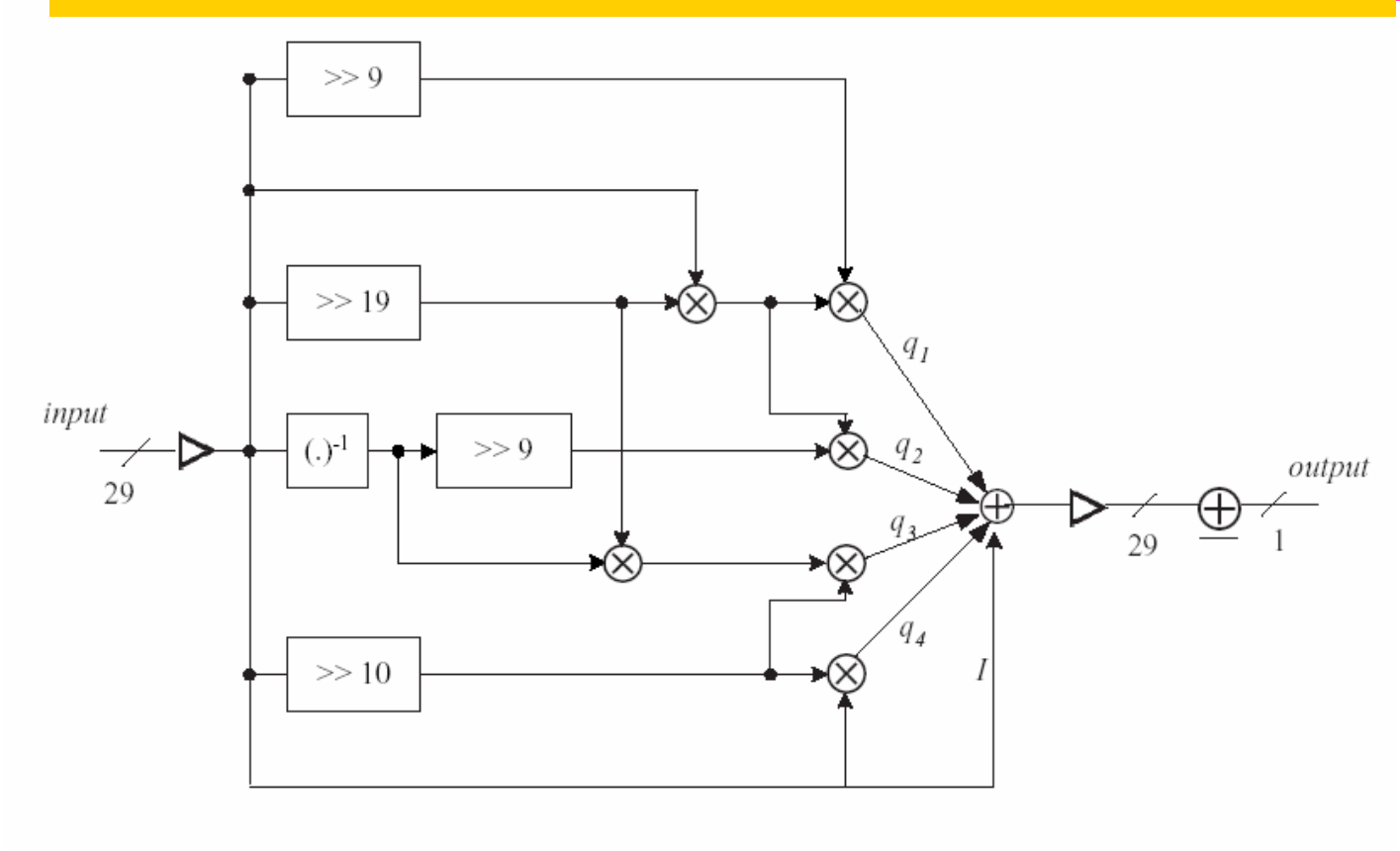



- Exponentiations in GF($2^{29}$)
- Expensive for polynomial basis
- In normal basis raising an element to a power of 2 is by cyclic shift
- We use normal basis multipliers to implement WG transformation

# Facts about WG Transformation Sequences

- WG transformation sequences was discovered in 1997 by Gong, Golomb and Gaal, which have 2-level autocorrelation, and verified their result up to period $2^{23}$-1.
- A few month later, No. etc. found another representation of WG and also verified their result for the same period as above.
- In 1999, Dillon proved the result for odd case.
- In 2005, Dobbertin and Dillon proved the result for even cases in their milestone work, which proved all conjectured 2-level autocorrelation sequences.
- In 2003, Gong and Youssef showed cryptographic properties of WG transformations.

# WG Transformation Operations under the Normal Basis




⊗ Normal Basis finite field multiplier over GF($2^{29}$)

⊕ Bit wise addition

⊕ Modulo 2 addition of 29 input bits (29 bit XOR gate)

# Security of the WG Cipher

- Randomness Properties of Keystream
  - Period is $2^{319}-1$
  - Balanced
  - 2-level autocorrelation
  - Ideal $t$-tuple distribution ($1 \leq t \leq 11$)
  - Linear complexity $\approx 2^{45.04}$
- Cryptographic Properties of WG Transformation
  - 1-order resilient
  - Algebraic degree 11
  - Nonlinearity $= 2^{28} - 2^{14} = 268419072$
  - Additive autocorrelation between $f(x+a)$ and $f(x)$ has three values: $0, \pm 2^{15}$.
  - 1-order propagation

University of Waterloo

# Security against Known Attacks

- Time/Memory/Data Tradeoff Attacks:
  - Size of internal state : $2^{319}$
- Algebraic Attacks
  - No of linear equations ≈ $\binom{319}{11}$
  - Attack complexity ≈$2^{182}$
- Correlation Attacks
  - WG transformation 1-order resilient
  - Nonlinearity very high $2^{28}$-$2^{14}$

# Design Rationale: Why WG?

**Set 1: Functions from the all known 2-level autocorrelation sequences**

- M-sequences
- GMW (including generalized GMW) sequences
- WG sequences
- Hyperoval sequences
- Dobbertin and Dillon's Kasami power function sequences
- Quadratic sequences and Hall sixtec residue sequences

**Set 2: Functions in Set 1 with nonlinearity $2^{n-1} - 2^{(n-1)/2}$ and 1-order resiliency**

- m-sequences with Kasami, Welch, and Niho decimations
- subset of GMW sequences
- WG sequences
- Glynn Type 1 hyperoval sequences
- Kasami power function sequences

**Set 3: Functions in Set 2 with three valued additive autocorrelation**

- m-sequences with Kasami decimation
- WG sequences
- Kasami power function sequences: for $k = 3$ and 5-term sequences

**Set 4: Functions in Set 3 with 1-order propagation**

- WG sequences
- 5-term sequences

Remark: Except for WG, there exist no other boolean functions that have all properties listed for Sets 2-4 without requiring 2-level autocorrelation.

# Design Rationale: Parameters Selection

- GF($2^{29}$) selected because:
  - Reasonable hardware complexity
  - Optimal normal basis exists
    - Minimize hardware complexity of multiplier
  - WG transformation exists
- 11 stage LFSR over GF($2^{29}$) selected:
  - Large internal state
  - Large linear complexity

University of Waterloo

# Hardware Implementation

- For WG, we can use 2 parallel normal basis multipliers over GF($2^{29}$)
- A typical parallel normal basis multiplier over GF($2^{29}$):
  - 841 AND gates and 1653 XOR gates
- 8 clock cycles for WG transformation
- For 1GHz clock, 125 Mbps throughput

University of Waterloo

# Summary of WG Cipher

The WG stream cipher

- Guaranteed keystream randomness properties, which is the only one with these properties in ECRYPT submissions
- Secure against time/memory/data tradeoff attacks, algebraic and correlation attacks
- Can be implemented in hardware with reasonable complexity