

Design of Pseudo-random Sequence Generators (Cont.)

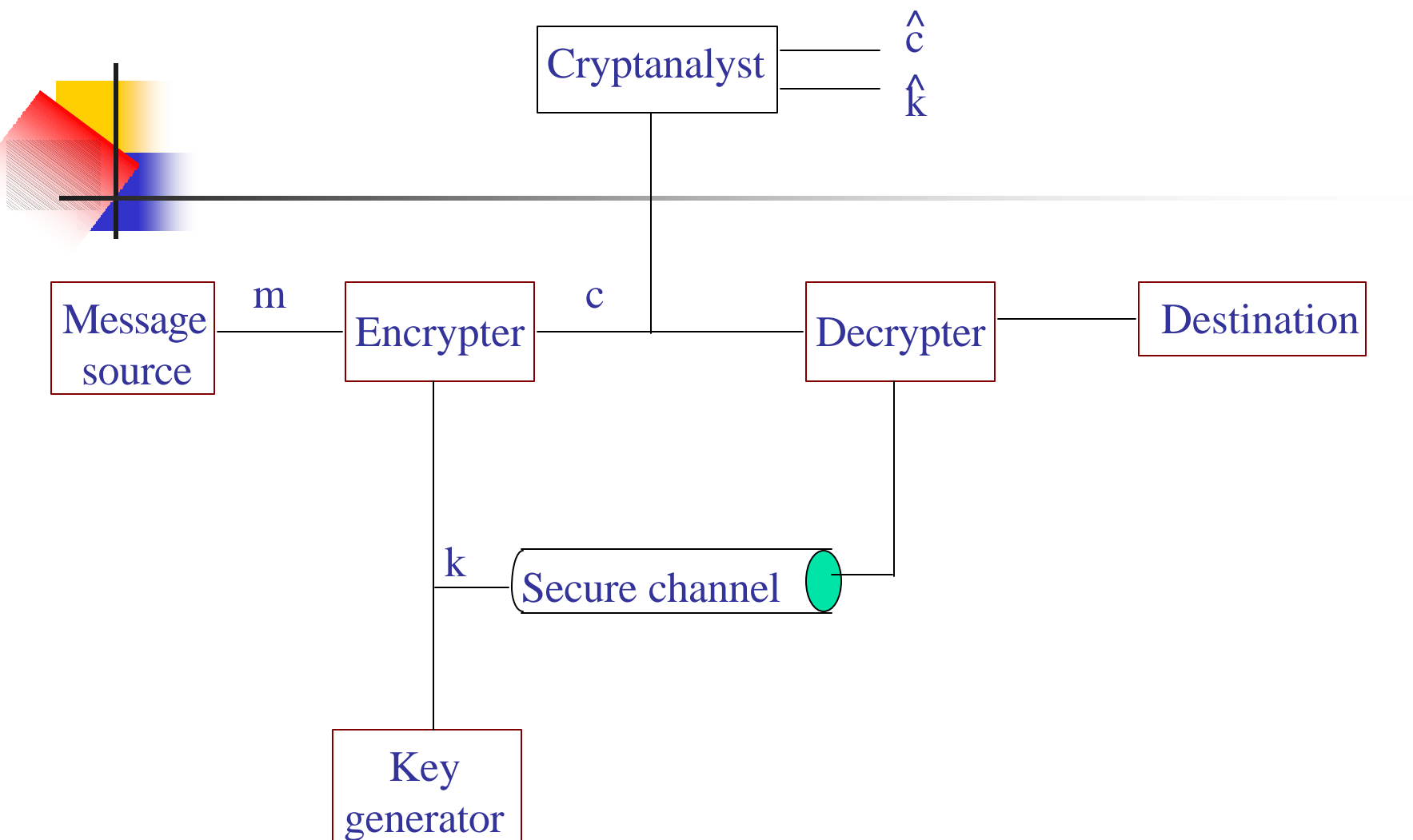
Lecture 2:

- **Design of PRSGs Towards Large Linear Span**
- **Applications in Stream Cipher Design: A5 and w7**



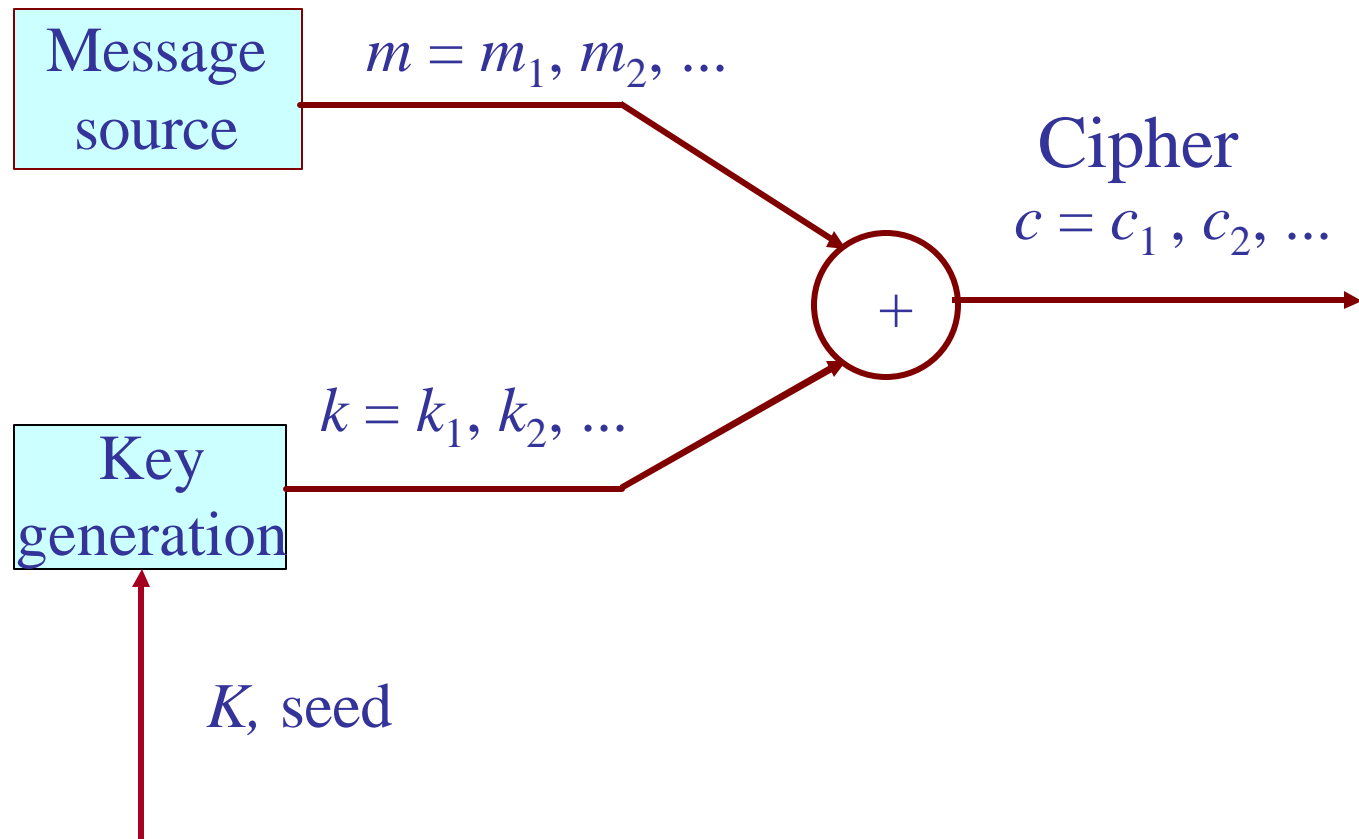
3. Design of PRSGs Towards Large Linear Span

- **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

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

- Shannon's result (1948): One-time-pad is unbreakable (requires long period of key stream).
- Massey's result (1969): If a binary sequence has linear span n then the entire sequence can be reconstructed from known $2n$ consecutive bits by the Berlekamp-Massey algorithm (requires large linear span of key stream).

Randomness Measurements (in Lecture 1)

Randomness Measurements for PRSG:

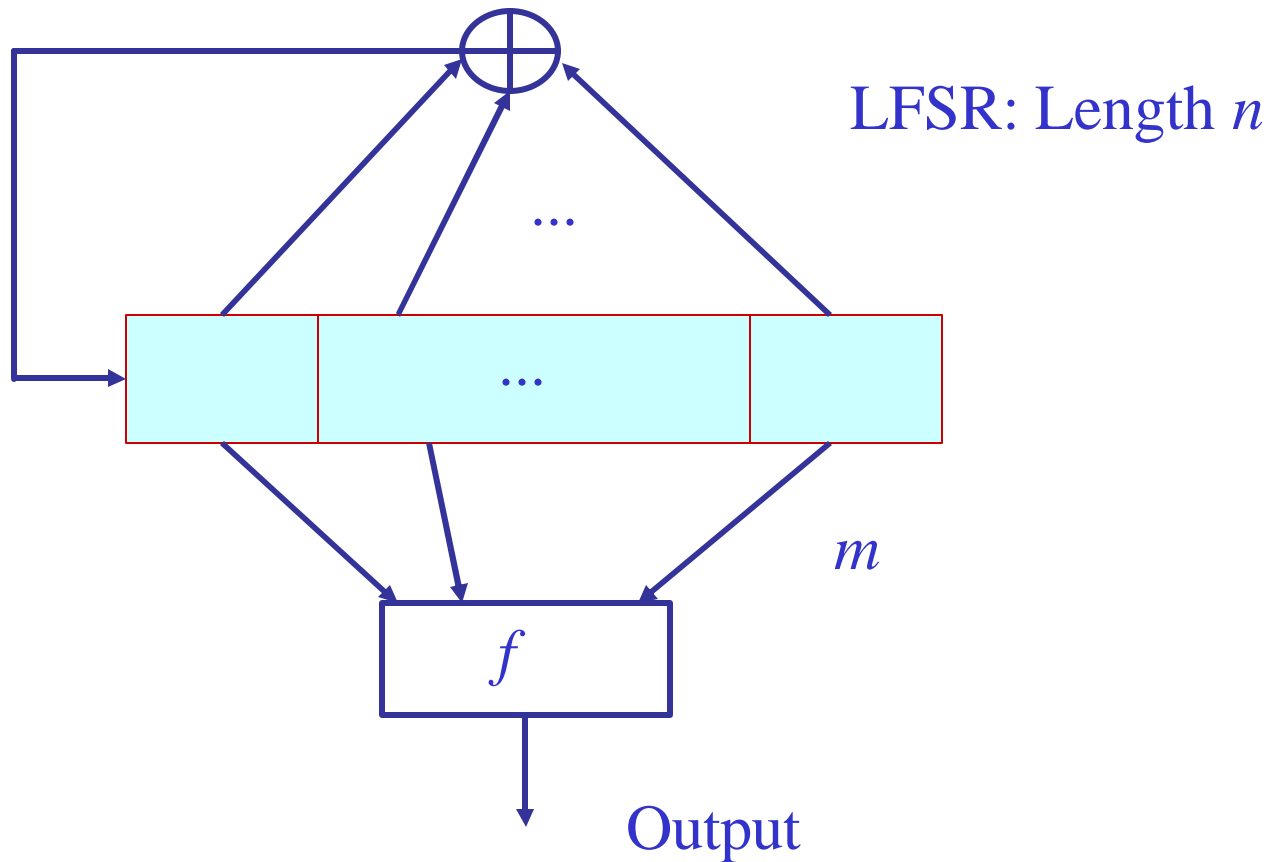
- Long Period
- Balance Property
- Run Property
- n -tuple Distribution
- Two-level Auto Correlation and Low Cross Correlation
- Large Linear Span



Known Approaches for Design of PRSG Based on LFSRs

- 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-Krawczyk-Mansour, 1993)

A Filtering Generator



A. Filtering Generators

Construction of A Filtering Generator (Key: 1973)

Parameters:

- Select $\underline{a} = \{a_i\}$ as an m -sequence over $GF(2)$ of degree n , or the characteristic polynomial of the LFSR is primitive.
- Select a positive integer m with $m \leq n$.
- Select m positive integers with $0 \leq d_1 < d_2 < \dots < d_m < n$ as the tap positions.
- Select a boolean function $f(x_1, x_2, \dots, x_m)$ in m variables x_1, \dots, x_m .

A sequence $\underline{s} = \{s_i\}$ whose terms are given by

$$s_i = f(a_{d_1+i}, a_{d_2+i}, \dots, a_{d_m+i}), i = 0, 1, \dots$$

is called a filtering sequence.

Profile of Filtering Sequences

1. Period 2^n-1 .

2. Linear Span, LS , has an upper bound

$$LS \leq \sum_{k=1}^m \binom{n}{k}$$

In particular, if $m = 2$, then $LS = n^2$.

3. It is not clear for the other randomness properties.

4. For some choices of the tap positions, the linear span will be lower bounded by the number of n choose m , $\binom{n}{m}$.



B. Combinatorial Function Generators

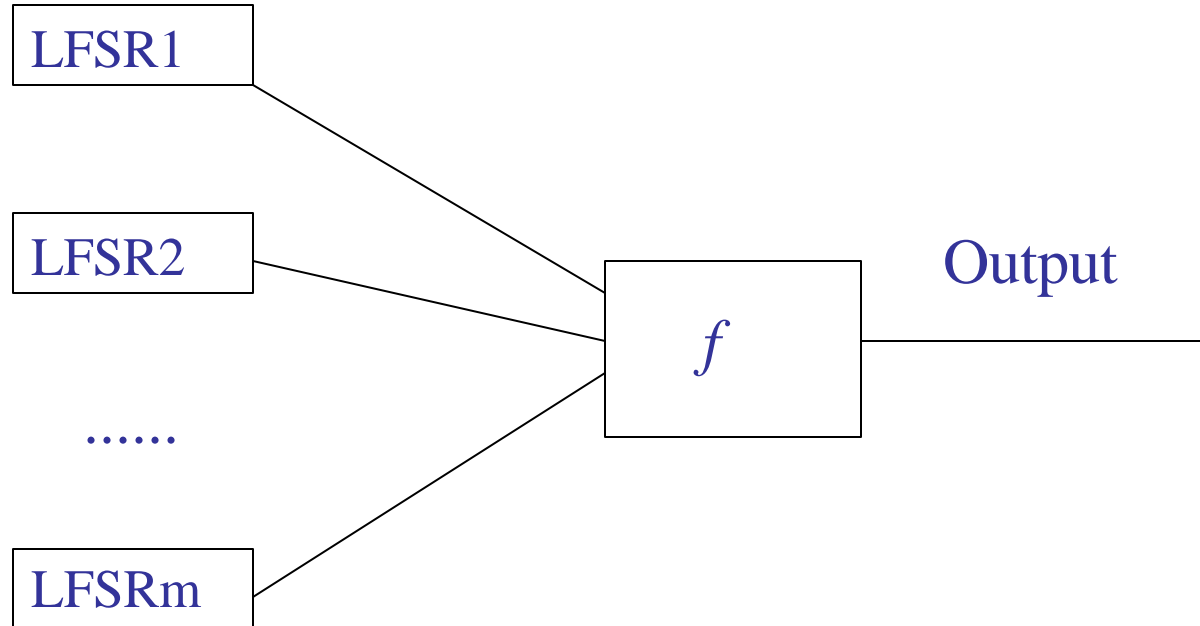
Construction of A Combinatorial Function Generator (Groth: 1971)

In the filtering generator, replacing $L^{d_j} \underline{a} = \{a_{d_j+i}\}_{i \geq 0}$ by a m -sequence $\underline{a}_j = \{a_{j,i}\}_{i \geq 0}$ of degree n_j with period

$$2^{n_j} - 1, j = 1, \dots, m,$$

the resulting sequence is called a combinatorial function sequence.

A Combinatorial Function Generator



Profile of Combinational Sequences

1. Period, Per , is given by

$$Per \mid \prod_{j=1}^m (2^{n_j} - 1)$$

with equality if n_1, \dots, n_m are pairwise coprime.

2. It is not clear for all other randomness properties for a general f .

Requirement for choice of f , a boolean function in m variables, in order to resistance to correlation attack (Siegenthaler, 1984), f should be balanced, has low correlation from all affine function (so-called nonlinearity) and correlation immunity (many tricks for selection of f !).

C. Clock-control Generators

Basic idea: change the clock pulse in the LFSRs.

Input: two sequences $\underline{\mathbf{a}} = \{a(t)\}$ and $\underline{\mathbf{b}} = \{b_i\}$, generated by LFSR1 and LFSR 2 respectively.

A simple model of clock-control generator: stop-and-go generator

Output sequence: $\underline{\mathbf{u}} = \{u(t)\}$

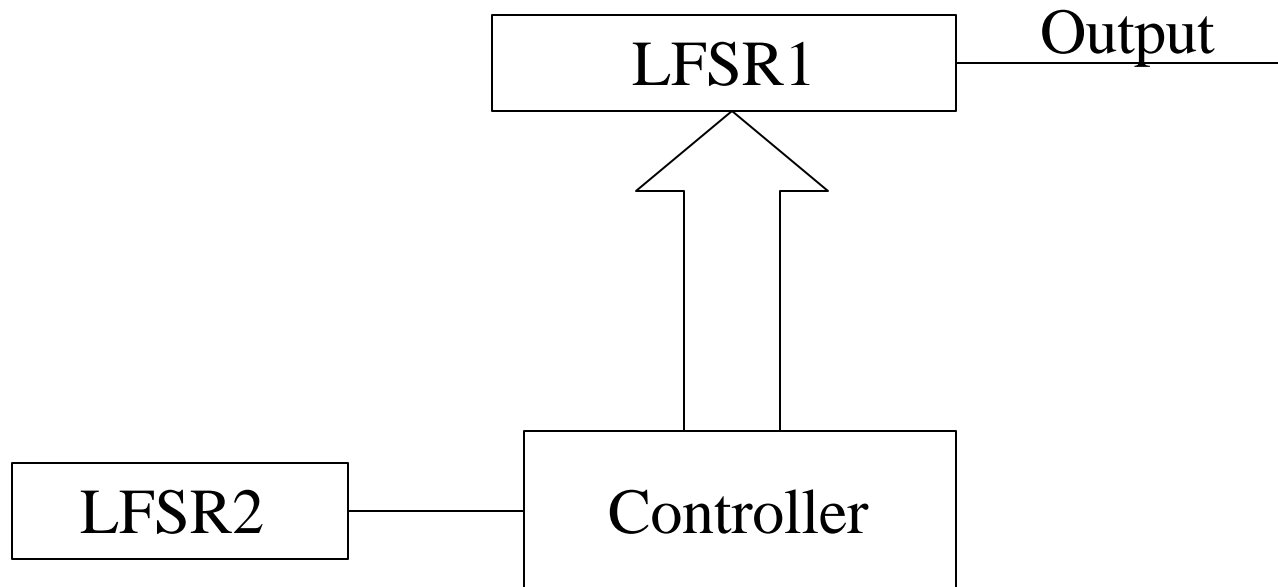
$$u(t) = a\left(\sum_{i=0}^t b_i\right), t = 0, 1, \dots$$

In other words, suppose that the previous bit is $u(t-1) = a(i_t - 1)$,
if $b_t = 1$, then the generator outputs $a(i_t)$: $u(t) = a(i_t)$.

Otherwise, the generator repeats the previous output bit:

$$u(t) = a(i_t - 1).$$

A Clock Controlled Generator



For example, if

$\underline{\mathbf{a}} = (a(0), a(1), \dots, a(6))$,

$\underline{\mathbf{b}} = 1001011$

then

$$u(0) = a(b_0) = a(1)$$

$$u(1) = a(b_0 + b_1) = a(1)$$

$$u(2) = a(b_0 + b_1 + b_2) = a(1)$$

$$u(3) = a(b_0 + b_1 + b_2 + b_3) = a(2)$$

$$u(4) = a(b_0 + b_1 + b_2 + b_3 + b_4) = a(2)$$

$$u(5) = a(b_0 + b_1 + b_2 + b_3 + b_4 + b_5) = a(3)$$

$$u(6) = a(b_0 + b_1 + b_2 + b_3 + b_4 + b_5 + b_6) = a(4)$$

$$u(7) = a(b_0 + b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_0) = a(5)$$

$$u(8) = a(b_0 + b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + b_0 + b_1) = a(5)$$

D. Shrinking generator

Input: \underline{a} and \underline{b} , the same as Stop-and-Go

Output: a sequence $\underline{u} = \{u(t)\}$.

Procedure:

Let the previous output be $u(i - 1)$ where $i = \sum_{j=0}^{t-1} b_j$

with the initial state

$u(0)=a(s)$ where $b_0 = b_1 = \dots = b_{s-1} = 0$ and $b_s=1$.

If $b_t = 1$, then the generator outputs $a(t)$: $u(i)=a(t)$, $i > 0$.

Otherwise, the generator discards $a(t)$.

For example, if

$\underline{a} = (a(0), a(1), \dots, a(6))$ and $\underline{b} = 1001011$,

then

$\underline{u} = (a(0), a(3), a(5), a(6)).$

4. Applications in Stream Cipher Design: A5 and w7

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

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

Note 1. A GSM conversation is sent as a sequence of frames per 4.6 millisecond, and each frame contains 228 bits.

Note 2. In A5 series, A5/2, similar as A5/1, is a weaker version than A5/1, and A5/3 is still in discussion in the work group of wireless communications.



Construction of A5/1 Generator:

Parameters:

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

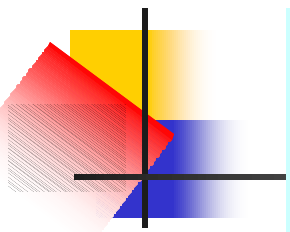
1. LFSR 1: $f_1(x) = x^{19} + x^5 + x^2 + x + 1$ generates $\underline{\mathbf{a}} = \{a(t)\}$.

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

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

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	0	0
	1	1	1
$(1,1,0)$	0	0	1
	1	1	0
$(0,1,1)$	0	1	1
	1	0	0
$(1,0,1)$	1	0	1
	0	1	0

Output:

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

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

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

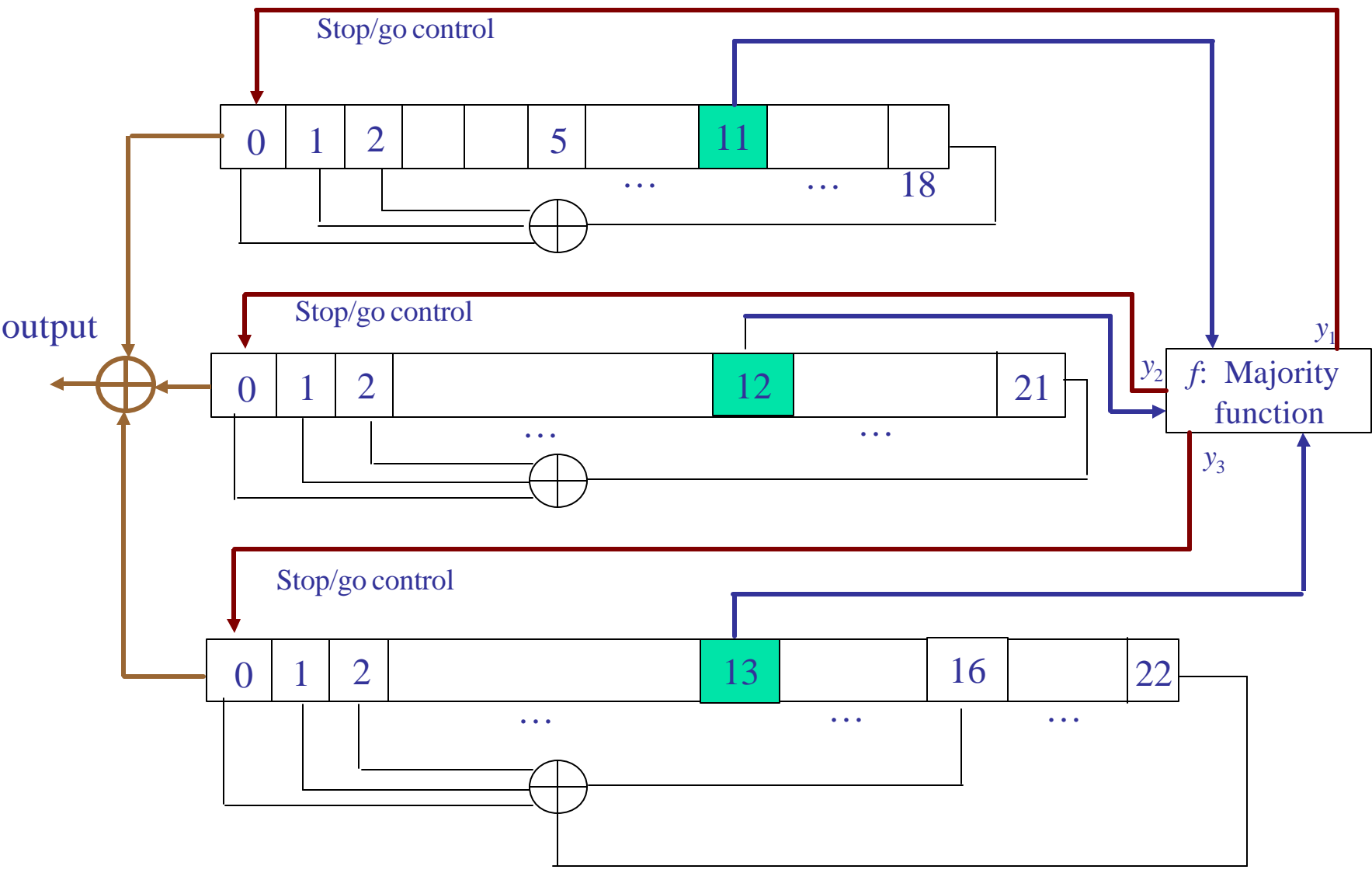


Figure 2: The A5/1 Stream cipher



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.

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 experiment shows that the period of A5/1 is around

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

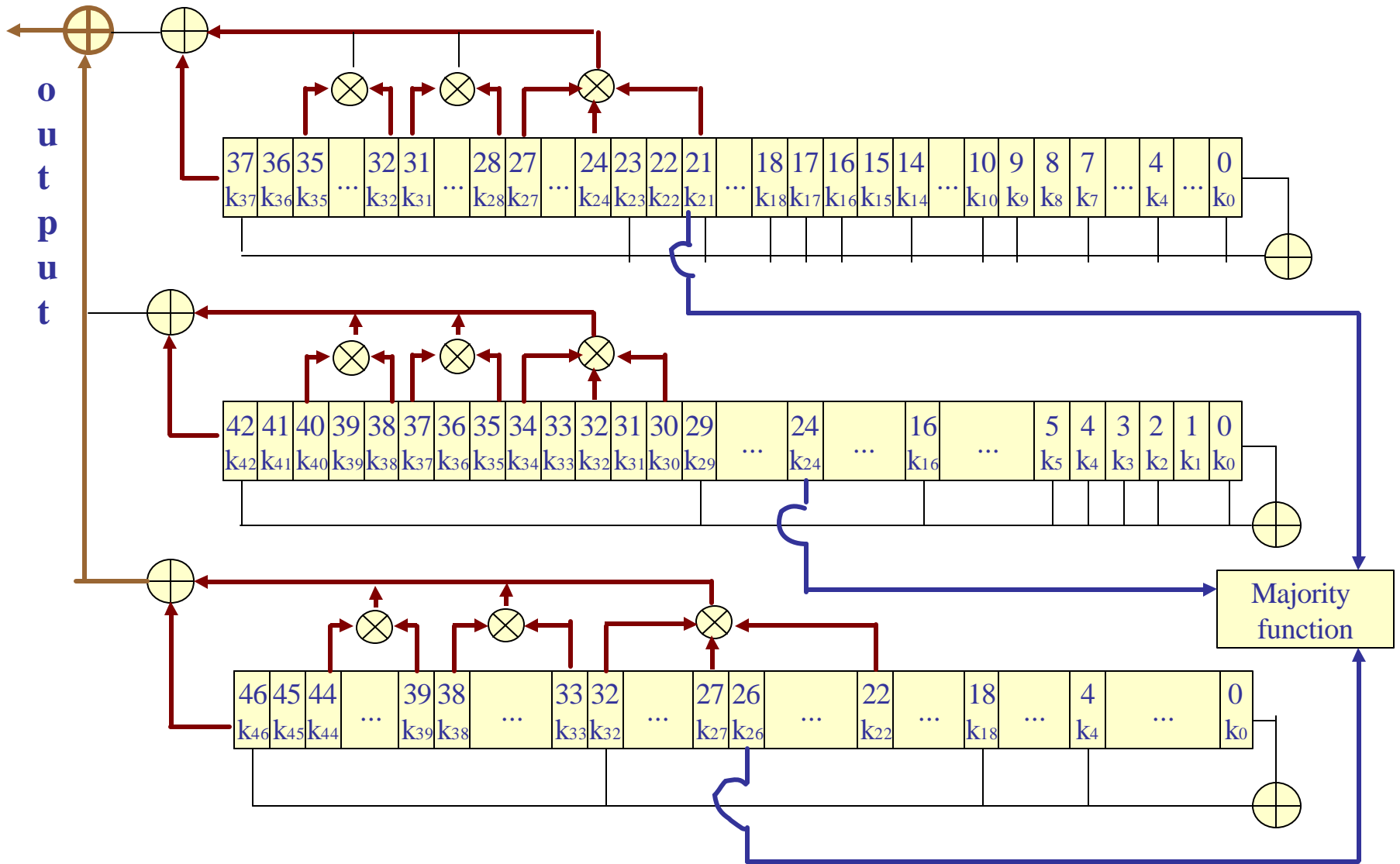
- Collision problem: different seeds (i.e., different initial states of three LFSRs) may result in the same key stream (our new results shows that only 70% seeds produce different key streams.)
- The majority function is the worst function in terms of correlation with all affine functions.



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 where the second one is illustrated as follows.



The W7 Cipher Algorithm