CS557: Cryptography

Stream Cipher

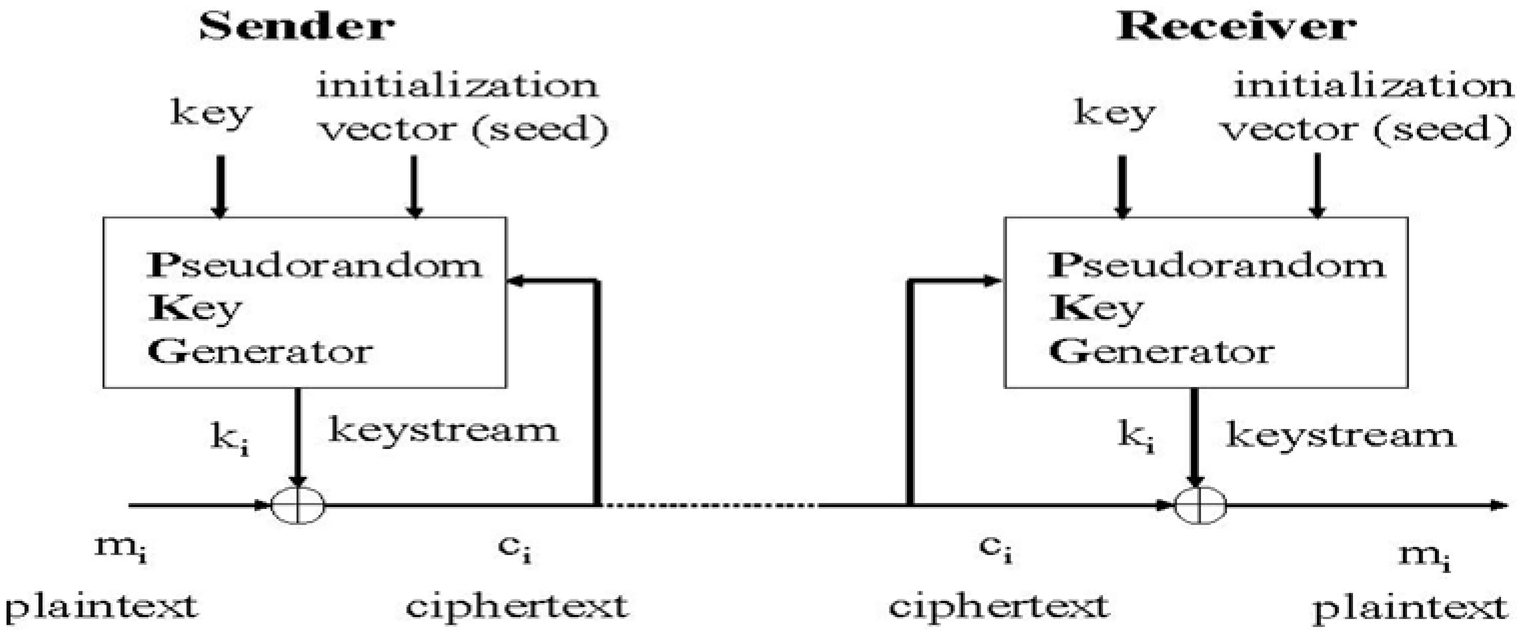
S. Tripathy
IIT Patna

Midsem Review

- Midsem Q
- MidsemKey
- Cryptography
  - Stream Ciphers
    - A5/1 (LFSR based)
    - RC4
  - Pseudo Random Number Generator

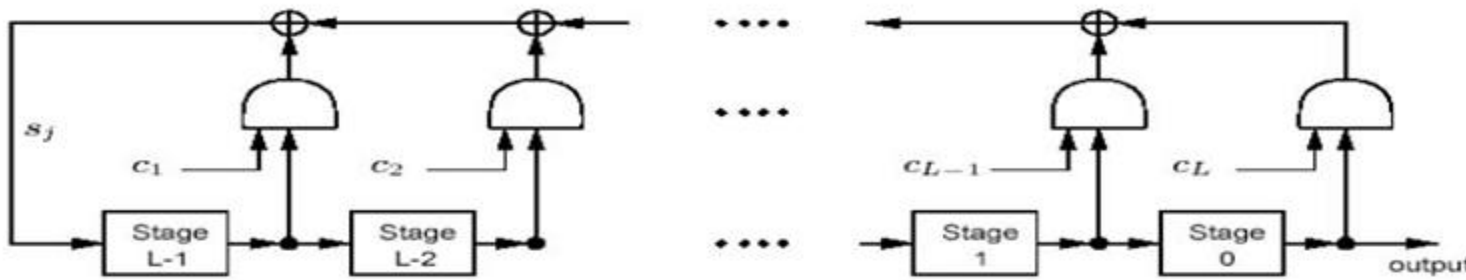
  - Term Project Title submission Reminder

Typical Stream Cipher



Main objective of a stream cipher construction is to get K as much random as possible.

Linear Feedback Shift Register (LFSR)

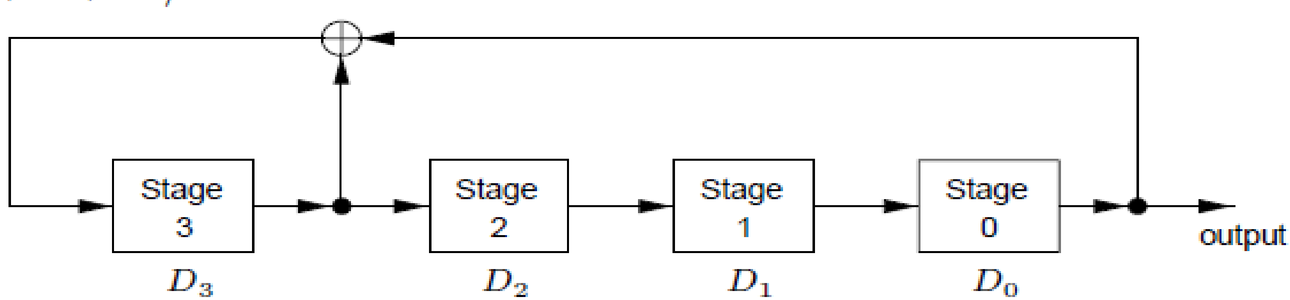


$\langle$ L, C(D) $\rangle$

Length **Connection polynomial, C(D)**

$$C(D) = 1 + c_1 D + c_2 D^2 + \ldots + c_L D^L$$

EX.: $LFSR\,\langle 4, 1 + D + D^4 \rangle$



LFSR $(4, 1+D+D^4)$

t	D_3	D_2	D_1	D_0
0	0	1	1	0
1	0	0	1	1
2	1	0	0	1
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	1	0	0	0
7	1	1	0	0

t	D_3	D_2	D_1	D_0
8	1	1	1	0
9	1	1	1	1
10	0	1	1	1
11	1	0	1	1
12	0	1	0	1
13	1	0	1	0
14	1	1	0	1
15	0	1	1	0

$$s = 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, \ldots,$$

Verify Random Property using Golomb's Postulates

Golomb's Randomness Postulates

R-1 : In every period, the number of 1's differ from the number of 0's by atmost 1.

R-2 : In every period, half the runs have length 1, 1/4th have length 2, 1/8th have length 3, etc., as long as the number of runs so indicted exceeds 1. Moreover, for each of the sequence lengths, there are (almost) equally many runs of 0's and of 1's

R-3 The auto correlation function C(ζ) is defined as

$$C(\tau) = \sum_{i=0}^{N-1} (-1)^{S_i + S_{i+\tau}}$$

is a 2-valued function

$$C(\tau) = \begin{cases} N & \text{if } \tau \equiv 0 \pmod{\mathrm{N}} \\ T & \text{if } \tau \neq 0 \pmod{\mathrm{N}} \end{cases}$$

Where T is a constant.

Linear feedback shift registers (LFSRs)

- LFSRs are used in many of the keystream generators
  - Well-suited to hardware implementation;
  - can produce sequences of large period
  - Can produce sequences with good statistical properties
  - can be readily analyzed using algebraic techniques.

Periodicity of the LFSR sequences •

- Periodicity of the LFSR sequences:
  - For some polynomials all the cycle lengths are equal to 2^L-1.
    - These polynomials are called primitive polynomials.
  - The sequence is then called m-sequence.
  - It has good statistical properties.
  - Example: 1+D+D4 is also primitive and thus we obtained a maximum length LFSR

Cryptanalysis of LFSR

- Vulnerable to known-plaintext attack
  - A LFSR can be described as
    $z_{m+i} = \sum_{j=0}^{m-1} c_j \, z_{i+j} \bmod 2$
  - Knowing 2*m* output bits, one can
    - construct *m* linear equations with *m* unknown variables $c_0, \ldots, c_{m-1}$
    - recover $c_0, \ldots, c_{m-1}$

Cryptanalysis of LFSR

- Given a 4-stage LFSR, we know
  - $z_4 = z_3c_3 + z_2c_2 + z_1c_1 + z_0c_0$ mod 2
  - $z_5 = z_4c_3 + z_3c_2 + z_2c_1 + z_1c_0$ mod 2
  - $z_6 = z_5c_3 + z_4c_2 + z_3c_1 + z_2c_0$ mod 2
  - $z_7 = z_6c_3 + z_5c_2 + z_4c_1 + z_3c_0$ mod 2
- Knowing $z_0, z_1, \ldots, z_7$, one can compute c_0, c_1, c_2, c_4.
- In general, knowing 2n output bits, one can solve an n-stage LFSR

$$z_j = c_1 z_{j-1} + c_2 z_{j-2} + \cdots + c$$

LFSR based Stream Cipher

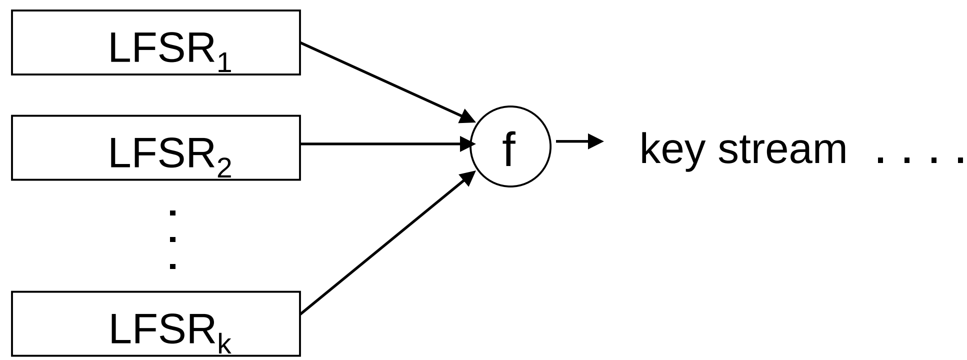
- LFSR-based stream ciphers can have some provable properties, like large period or linear complexity.

- Drawback for cryptography:
  - LFSRs easy to predict.
  - Solve a system of linear equations for unkonwn state bits and recursion coefficients, or use Berlekamp-Massey algorithm.

- Destroy linearity by
  - Nonlinear filter/combining functions on outputs of one or several LFSRs.
  - Use of output of one/several LFSRs to control the clock of one/more other LFSRs.

Achieve Non-linearity in LFSR

- (i) using a nonlinear combining function on the outputs of several LFSRs;

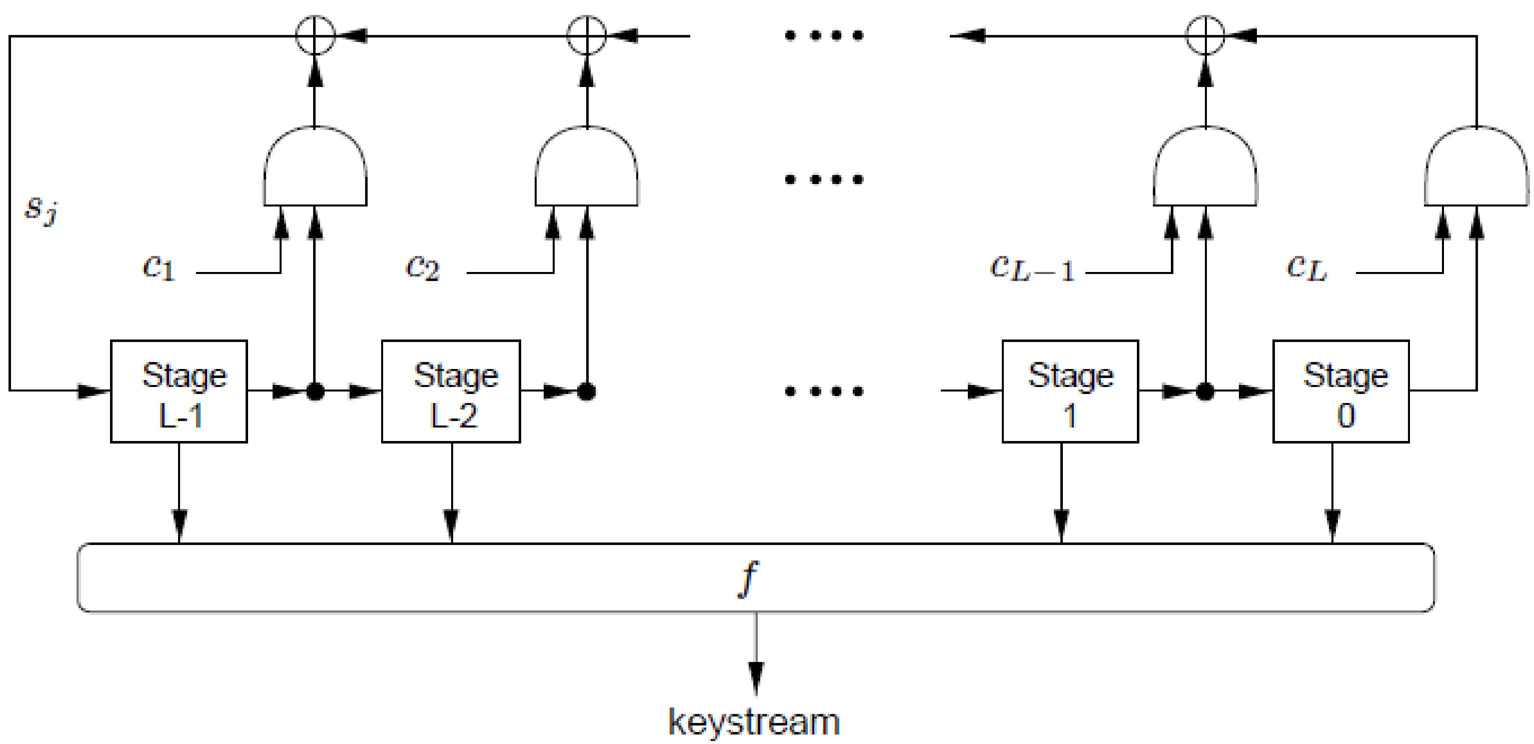


Desirable properties of f:

- high non-linearity
- long "cycle period" $(\sim 2^{n1+n2+\ldots+nk})$
- low correlation with the input bits

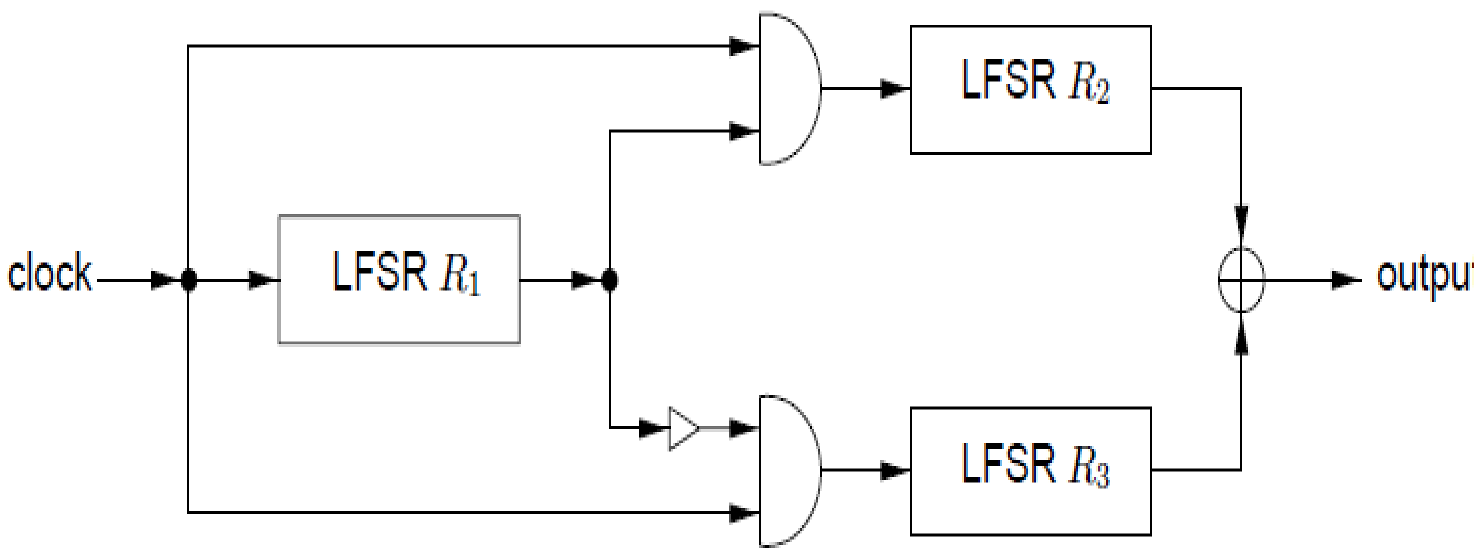
Achieve Non-linearity in LFSR-II

(ii) using a nonlinear filtering function on the contents of a single LFSR



Achieve Non-linearity in LFSR-III

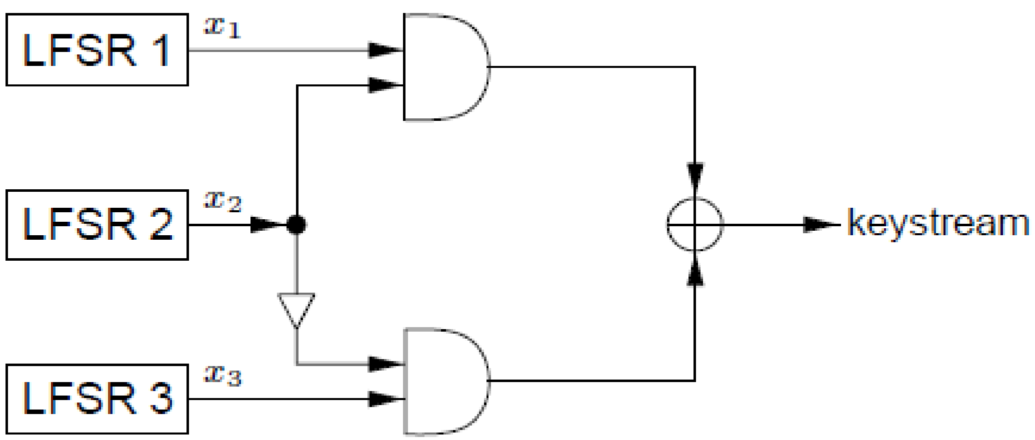
(iii) using the output of one (or more) LFSRs to control the clock of other(s)



Example LFSR-Based Ciphers

- Geffe Generator:
  - Three LFSRs
  - $LFSR_2$ is used to choose between $LFSR_1$ & $LFSR_3$:

$$Z = (x^{(1)} \wedge x^{(2)}) \oplus (\neg x^{(2)} \wedge x^{(3)})$$

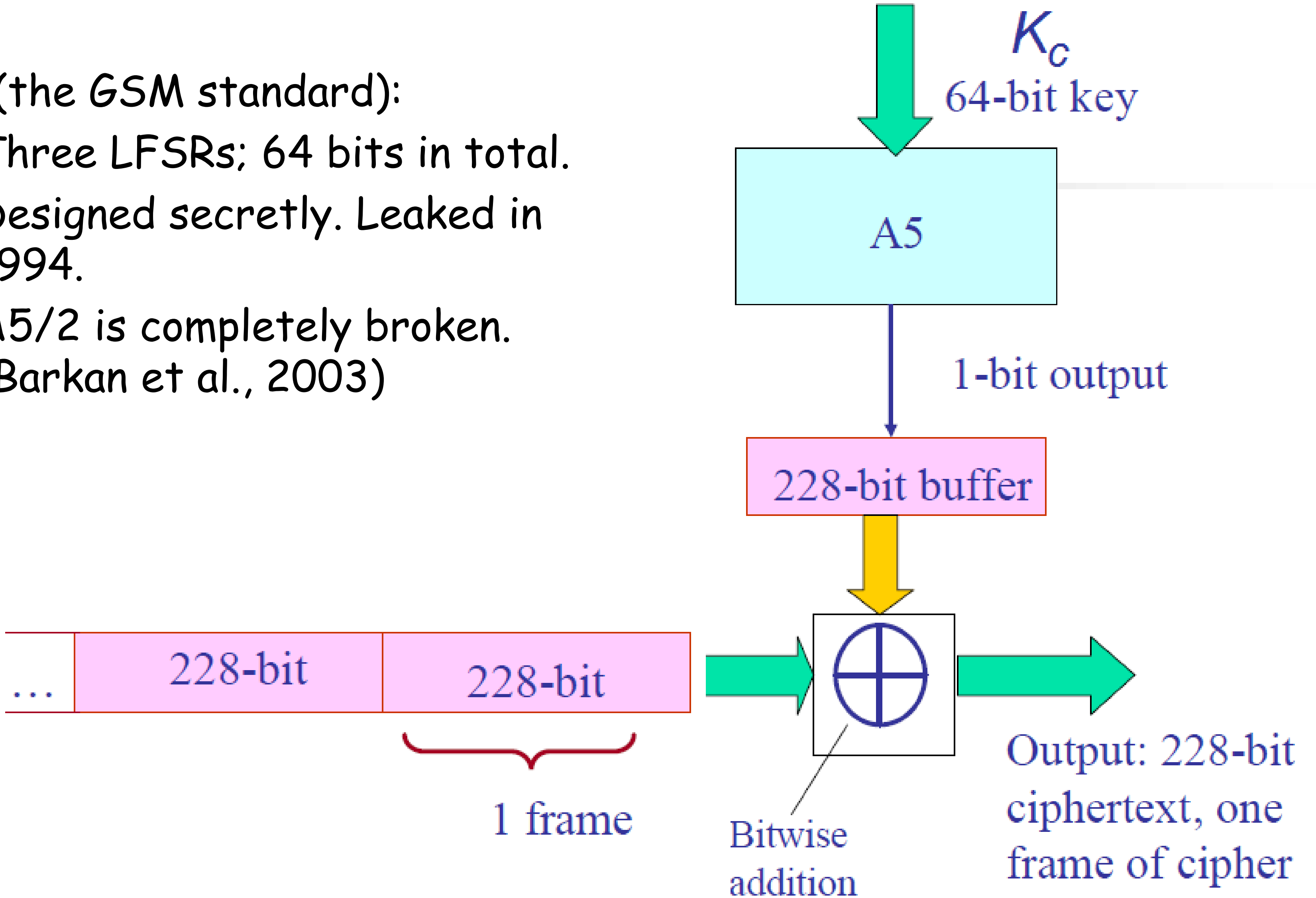


Security:

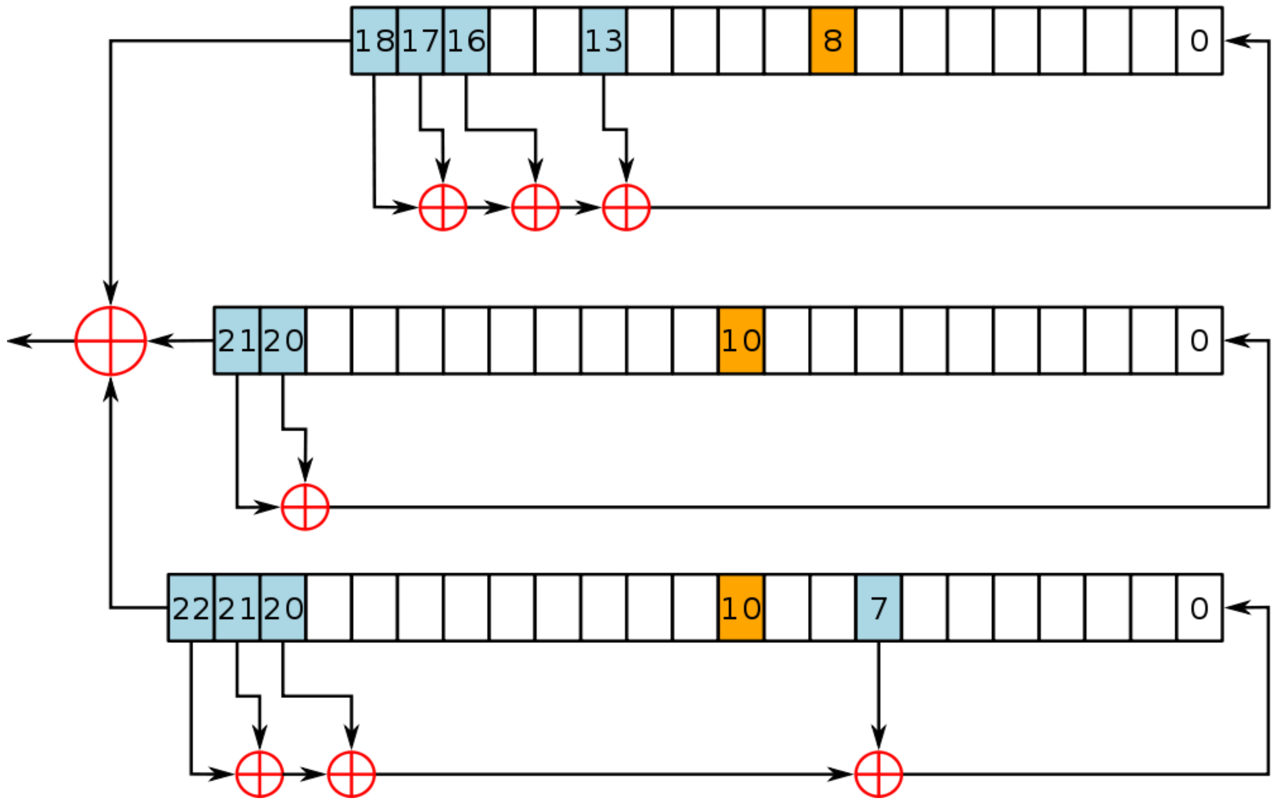
$$\begin{aligned} P(z(t) = x_1(t)) &= P(x_2(t) = 1) + P(x_2(t) = 0) \cdot P(x_3(t) = x_1(t)) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}. \end{aligned}$$

LFSR-Based Ciphers

- A5 (the GSM standard):
  - Three LFSRs; 64 bits in total.
  - Designed secretly. Leaked in 1994.
  - A5/2 is completely broken. (Barkan et al., 2003)



GSM A5/1



- The A5/1 stream cipher uses three LFSRs.
- A register is clocked if its clocking bit (orange) agrees with one or both of the clocking bits of the other two registers. (majority match)

A5/1 LFSRs

- 19 bits
  - $x^{19} + x^{18} + x^{17} + x^{14} + 1$
  - clock bit 8
  - tapped bits: 13, 16, 17, 18
- 22 bits
  - $x^{22} + x^{21} + 1$
  - clock bit 10
  - tapped bits 20, 21
- 23 bits
  - $x^{23} + x^{22} + x^{21} + x^{8} + 1$
  - clock bit 10
  - tapped bits 7, 20, 21, 22

- Least significant bit numbered 0
- Tapped bits of each LFSR are XORed to create value of next 0 bit.
- Output bits of the three LFSRs are XORed to form the keystream bit

A5/1

- Each cycle, look at the three clock bits. The majority value, c_m, is determined.
- In each LFSR, if the clock bit matches c_m, the registers are clocked.
- In each cycle, 2 or 3 LFSRs will be clocked.

A5/1 Initialization

- Registers set to all 0's
- Incorporate the key and frame number:
  - For 64 cycles, the key is mixed in by XORing the i^{th} key bit with the least significant bit of each register
  - For 22 cycles, the 22 bit frame value is mixed in – same as with key value
  - Normal clocking used
- 100 cycles are run using the majority clocking, the output is discarded
- End result is the initial state

Software-Oriented Stream Ciphers

- LFSRs slow in software
- Alternatives:
  - Block ciphers (or hash functions) in CFB, OFB, CTR modes.
  - Stream ciphers designed for software:
    - RC4
    - SEAL