Stream Cipher

❑Introduction

- Same P ^{1} Same C, memory, state cipher
- bit-by-bit Exor with pt and key stream
  ($c_i = m_i$ Å z_i)
- Encryption = Decryption --> Symmetric
- Use LFSR (Linear Feedback Shift Register)
- (external) Synchronous or self-synchronous
- Security measure : Period of key stream,
  LC(Linear Complexity)

Sequence

❑ Def)

– $s=s_0,s_1$, …: infinite seq.,

– *n* term of s : $s^n=s_0,s_1, ..,s_{n-1}$

– if $s_i = s_{i+n}$ for all *i* >=0, s is periodic seq. having period n.

– run : subsequence of consecutive ‘ 0’ (gap) or consecutive ‘ 1’ (block)

Golomb' s postulates(I)

s^N : periodic seq. of N

(1) for a cycle of s^N, 0~1 balanceness, i.e, #{s_i=1} - #{s_j=0} =1, for all *i* ¹ *j* < N.

(2) for a cycle of s^N, half the runs have length 1, 1/4 have the length 2, .., etc.

(3) Autocorrelation* function has 2 values.

$$N \bullet C(t) = \sum_{i=0}^{N-1} (2s_i - 1) \cdot (2s_{i+t} \text{ -}1) = \begin{Bmatrix} N, \text{ if } t = 0 \\ K, \text{ if } 1 \le t \le N-1 \end{Bmatrix}$$

* Measuring similarity between original and t-shifted sequences

** If satifies all, called as Pseudo-Noise(PN) sequence.

Golomb' s postulates(II)

(Ex) s^{15} = 0,1,1,0,0,1,0,0,0,1,1,1,1,0,1

(1) #{0} = 7, #{1}=8

(2) 8 runs, 4 runs with length 1 (2 gaps, 2 blocks), 2 runs with length 2 (1 gap, 1 block), 1 run with length 3 (1 gap), 1 run with length 4 (1 block)

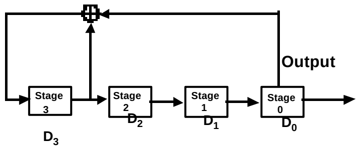
(3) Autocorrelation function, C(0)=1, C(t)= -1/15

Thus, PN-seq.

LFSR(I)

- Notation: < L, C[D]> where connection poly.
  C[D] = 1 + c_1D + c_2D^2 + ..+ c_LD^L Î Z_2[D]
- If c_L=1, {i.e., deg{C[D]}=L}, singular polynomial.
- If initial stage is [s_{L-1}, ..., s_1,s_0], output seq. s_0,s_1, ...s_j = (c_1s_{j-1} + $c_2 s_{j-2}$ + ...+ $c_L s_{j-L}$) mod 2 , j ³ L
- (Ex) <4, 1 + D + D^4> , s_0 = [0,1,1,0]
- Finite State Machine

t	D_3	D_2	D_1	D_0	t	D_3	D_2	D_1	D_0
0	0	1	1	0 (6)	8	1	1	1	0 (14)
1	0	0	1	1 (3)	9	1	1	1	1 (15)
2	1	0	0	1 (9)	10	0	1	1	1 (7)
3	0	1	0	0 (4)	11	1	0	1	1 (11)
4	0	0	1	0 (2)	12	0	1	0	1 (5)
5	0	0	0	1 (1)	13	1	0	1	0 (10)
6	1	0	0	0 (8)	14	1	1	0	1 (13)
7	1	1	0	0 (12)	15	0	1	1	0 (6)



Output seq. = 0,1,1,0,0,1,0,0,0,1,1,1,1,0,1,0

LFSR(II)

m	k(k_1,k_2,k_3)	m	k(k_1,k_2,k_3)	m	k(k_1,k_2,k_3)	m	k(k_1,k_2,k_3)
2	1	12	7,4,3	22	1	32	28,27,1
3	1	13	4,3,1	23	5	33	13
4	1	14	12,11,1	24	4,3,1	34	15,14,1
5	2	15	1	25	3	35	2
6	1	16	5,3,2	26	8,7,1	36	11
7	1	17	3	27	8,7,1	37	12,10,2
8	6,5,1	18	7	28	3	38	6,5,1
9	4	19	6,5,1	29	2	39	4
10	3	20	3	30	16,15,1	40	21,19,2
11	2	21	2	31	3	41	3

* Primitive polynomial over Z_2: x^m+x^k+1(trinomial) or x^m + x^{k1}+x^{k2}+x^{k3}+1(pentanomial) called as Maximum-length Shift Register Seq., M –seq.

* # of monic primitive poly =f(2^m-1)/m in Z_2[x] where f is Euler-phi ft.

LFSR(III)

- ❑ Well suited for H/W implementation
- ❑ Produce seq. of large period
- ❑ Good statistical properties
- ❑ Readily analyzed by algebraic structure
- ❑ Breakable by consecutive 2 * L sequence : depends on computing an inverse matrix whose complexity is O(L^3), L : length of LFSR. -> one LFSR is useless.

Linear Complexity(I)

- ❑ (Def) LC of finite seq . s^n, L(s) : length of shortest LFSR that generates a seq. having s^n as its 1st n terms.

  : Berlekamp-Massey algorithm
- ❑ (Properties of LC) s,t : binary seq.
  - For any n ³ 1, 0 £ L(s^n) £ n
  - L(s^n) =0 iff s^n is ‘0’ seq. of length n.
  - L(s^n) =n iff s^n=0,0,..,0,1.
  - If s is periodic with period N, L(s^n) £ N.
  - L(sÅt) £ L(s) + L(t)

Linear Complexity(II)

- ❑ s^n : random seq. from all seq. of length n
- ❑ Expectation value of LC $E(L(s^n)) = \frac{n}{2} + \frac{4+B(n)}{18} - \frac{1}{2^n}\left(\frac{n}{3} + \frac{2}{9}\right)$

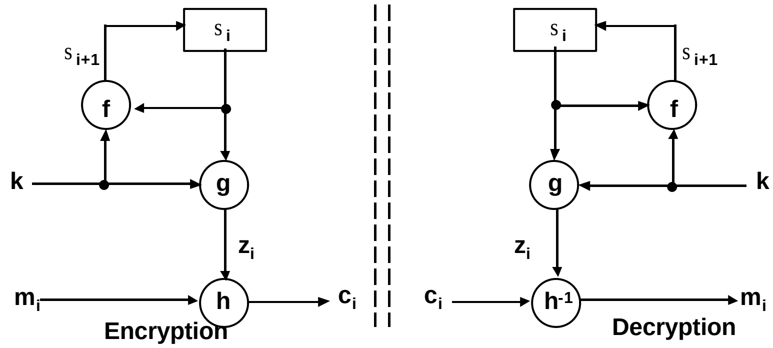
  where B(n)=0 if even n, otherwise 0

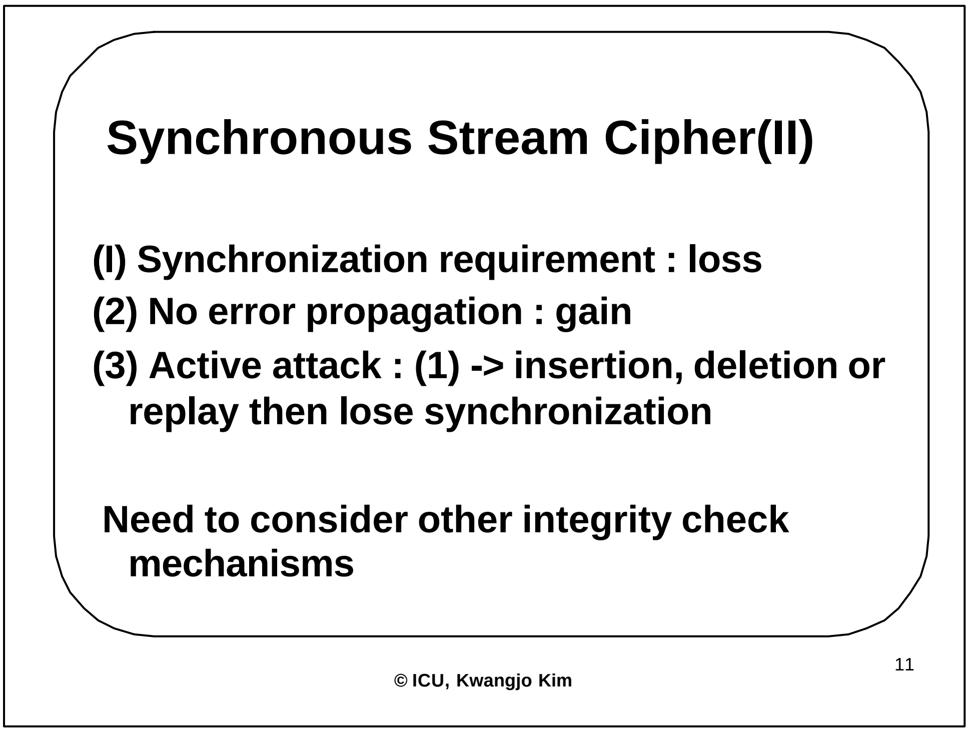
  for large n E(L(s^n)) » n/2 + 2/9 and Var(L(sn)) » 86/81
- ❑ (Def) LCP (Linear Complexity Profile)

L_N : LC of s^N=s0,s1, ..s_{N-1}, L_1, L_2, ...L_N is LCP

Synchronous Stream Cipher(I)

- f : next state ft, s_{i+1}= f(s_i , k), s_0 : initial value
- g : keystream generating ft, z_i = g (s_i , k), k : key
- h : output ft, c_i = h (z_i, m_i) , m_i : pt, z_i : key stream, c_i:ct





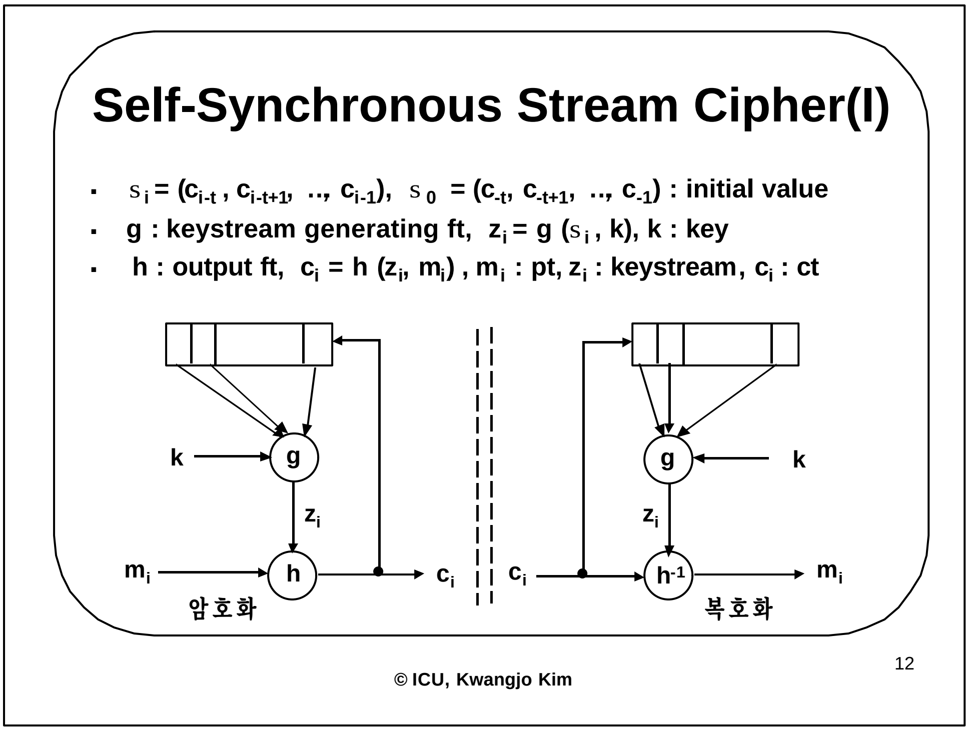
Synchronous Stream Cipher(II)

(l) Synchronization requirement : loss

(2) No error propagation : gain

(3) Active attack : (1) -> insertion, deletion or replay then lose synchronization

Need to consider other integrity check mechanisms



Self-Synchronous Stream Cipher(I)

- $s_i = (c_{i-t}, c_{i-t+1}, .., c_{i-1})$, $s_0 = (c_{-t}, c_{-t+1}, .., c_{-1})$: initial value
- g : keystream generating ft, $z_i = g(s_i, k)$, k : key
- h : output ft, $c_i = h(z_i, m_i)$, m_i : pt, z_i : keystream, c_i : ct

Self-Synchronous Stream Cipher(II)

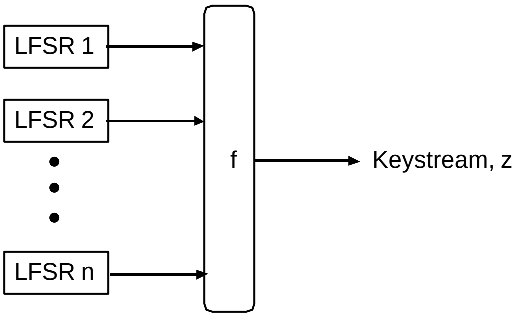
(1) self-synchronization : gain

(2) error propagation : loss

(3) active attack : (2) -> easy to find passive modification, (1) -> by active adv., difficult to find

Þ other integrity check mechanism required.

Nonlinear Combiner(I)

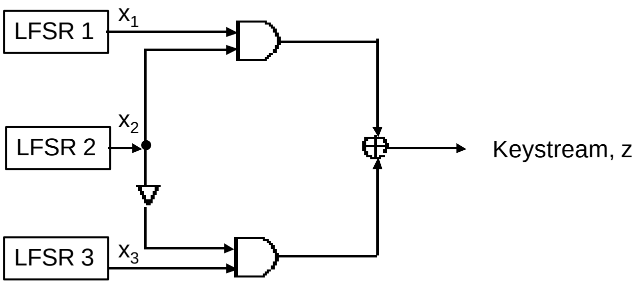


Algebraic Normal Form (ANF) : mod. 2 sum of distinct *m*-th order product of its variable, 0 <= m <= n

Ex) $f(x_1,x_2,x_3,x_4,x_5)=1 + x_2+ x_3 + x_4 + x_4x_5 + x_1x_2x_3x_4,$ deg(f) =4

Nonlinear Combiner(II)

❑ **Geffe generator**



$f(x_1,x_2,x_3) = x_1x_2 \oplus (1+x_2)x_3 = x_1x_2 \oplus x_2x_3 \oplus x_3$

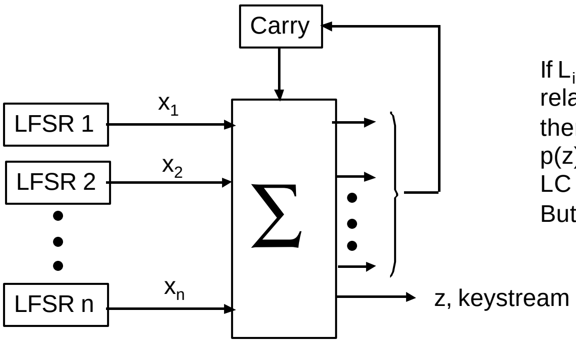
$p(z) : (2^{L1}-1)(2^{L2}-1)(2^{L3}-1)$

where L_1, L_2 and L_3 are relatively prime

$L(z) = L_1L_2 + L_2L_3 + L_3$: Correlation attack is possible !

Nonlinear Combiner(III)

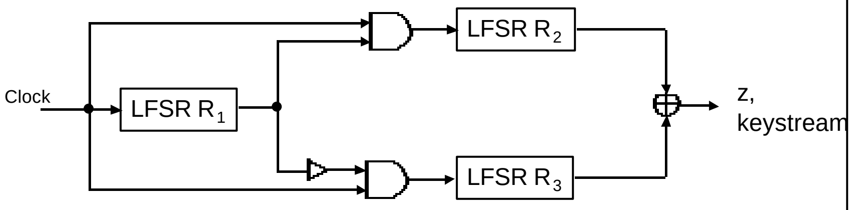
❑ **Summation generator**



If L_i and L_j are pairwise relatively prime,
then
$p(z) = \prod_{i=1}^{n} (2^{L_i}-1)$
LC ≈ p(z)
But weak in 2-adic span

Clock-controlled generator(I)

❑ **Alternating step generator**

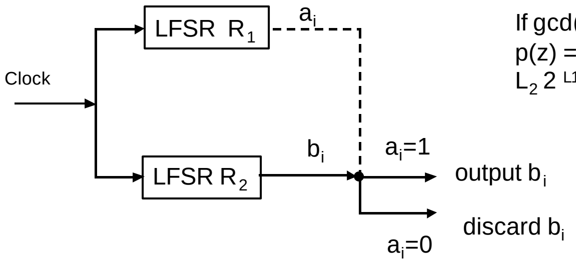


R_1 : de Brujin seq. of period 2^{L1} p(z) = $2^{L1}(2^{L2}-1)(2^{L3}-1)$

R_2,R_3 : m-seq s.t., gcd(L_1, L_2)=1 L(z) : $(L_2 + L_3)\ 2^{L1-1} < L(z) <= (L_2+L_3)\ 2L_1$

Clock-controlled generator(II)

❑ **Shrinking generator**

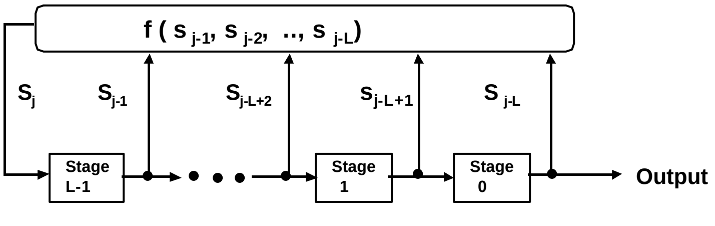


If gcd(L1, L2) =1,
p(z) = $(2^{L2}-1)\ 2^{L1-1}$
$L_2\ 2^{L1-2} < L(z) < L_2\ 2^{L1-1}$

Other generators

- ❑ Cascade Generator
- ❑ CSPRBG(Cryptographically Secure Pseudo Random Bit Generator)
  - – RSA LSB Generator
  - – BBS Generator
- ❑ Pseudo-noise Generator
  - – Noise Diode or Noise Transistor

Nonlinear FSR



f() : nonlinear ft

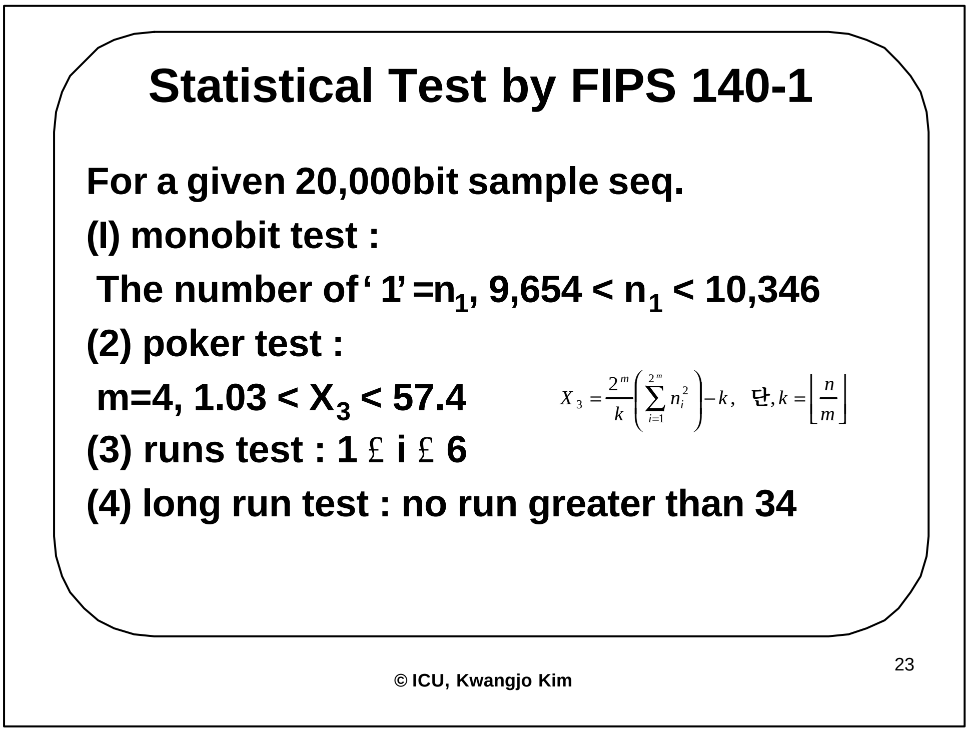
Security of Stream Cipher

- ❑ Period : Depends on req’ d level of security
- ❑ Linear Complexity
  - – shortest LFSR that generates a given seq.
- ❑ Measure against Correlation Attack
  - – Correlation Immune ft
  - – Nonlinear ft
- ❑ DC,LC, and DFA are applicable

* A5 crack survey : http://www.jya.com/crack-a5.htm

Statistical Randomness

- ❑ Frequency Test
- ❑ Serial Test
- ❑ Run Test
- ❑ Poker Test
- ❑ Spectral Test
- ❑ Linear Complexity Profile
- ❑ Quadratic Complexity



Statistical Test by FIPS 140-1

For a given 20,000bit sample seq.

(I) monobit test :

The number of ‘1’=n_1, 9,654 < n_1 < 10,346

(2) poker test :

m=4, 1.03 < X_3 < 57.4 $X_3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k,$ 단, $k = \left\lfloor \frac{n}{m} \right\rfloor$

(3) runs test : 1 £ i £ 6

(4) long run test : no run greater than 34

23

© ICU, Kwangjo Kim