Cryptographic Security

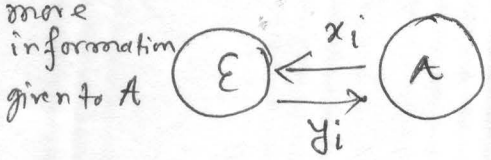
- Kerckhoff's Principle: Assume that the adversary knows the algm. that is used. The secret is only the secret key.

- Attack Models: (Passive Attacks - eavesdropping)

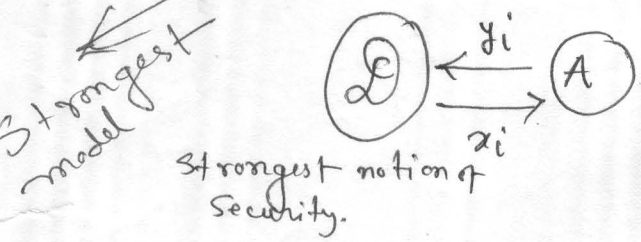
  - Ciphertext only attack: The opponent possesses a string of ciphertext, y.
    y

  - Known plaintext attack: The opponent possesses a string of plaintext, x, and the corresponding ciphertext, y.
    (x, y).

  - Chosen plaintext attack: The opponent has obtained temporary access to the encryption machinery. Hence he can choose a plaintext string, x, and construct the corresponding ciphertext string, y.
    $(x_1, y_1), \cdots, (x_n, y_n)$



  - Chosen ciphertext attack: The opponent has obtained temporary access to the decryption machinery. Hence he can choose a ciphertext string, y, and construct the corresponding plaintext string, x.



- Adversarial Goal: Key recovery, distinguishing attack, malleability, Other application specific security goals.

∴ Passive attack → only listen to the traffic
∴ Active attack → can alter messages
(replay, delete, modify, insert)

- Hill cipher → difficult to break with a ciphertext-only attack
  → easy to break with a known plaintext attack.

Affine fun → $ax+b$

$x \in \mathcal{P} = (Z_{26})^m$

$K \in \mathcal{K} = Z_{26}^{m \times m}$

$(y_1, y_2, \dots, y_m) = (x_1 x_2 \dots x_m) \begin{pmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{pmatrix}$

$\Rightarrow y_i$ = linear Combination of $x_1, x_2, \dots, x_m$ with coefficients $k_{1i}, k_{2i}, \dots, k_{mi}$ ($k_i \rightarrow$ i-th Column of K)

- $y = e_K(x) = xK$
- $d_K(y) = yK^{-1}$

$\Big\}$ mod 26

y, x known $\Rightarrow K = x^{-1}y$

0	1	2	3	4	5	6	7	8	9	10
a	b	c	d	e	f	g	h	i	j	k

11	12	13	14	15	16
l	m	n	o	p	q

17	18	19	20	21	22
r	s	t	u	v	w

23	24	25
x	y	z

Example: friday

↓ $m = 2$

P Q C F K U

Find a 2×2 matrix K over Z_{26} s.t.

$(15,16) = (5,17)K = e_K(5,17)$

$(2,5) = (8,3)K = e_K(8,3)$

$(10,20) = (0,24)K = e_K(0,24)$

$\Rightarrow \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} K$

$9^{-1} = 3 \bmod 26$

$121 \equiv 17 \bmod 26$, $54 \equiv 2 \bmod 26$, $7 \equiv 7 \bmod 26$

$15 - 136 = -121 \bmod 26 = -17 = 9$

$K = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix}$

$= 9 \begin{pmatrix} 3 & 1 \\ 8 & 15 \end{pmatrix} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 10 \\ 8 & 3 \end{pmatrix}$

$\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} = 9^{-1} \begin{pmatrix} 3 & -17 \\ -8 & 5 \end{pmatrix}$

$= 3 \begin{pmatrix} 3 & 9 \\ 18 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}$

(3)

- Defn $m \times m$ matrix

$$\begin{cases} x_j = (x_{1,j}, x_{2,j}, \cdots, x_{m,j}) \\ y_j = (y_{1,j}, y_{2,j}, \cdots, y_{m,j}) \end{cases} \longrightarrow m \text{ distinct plaintext-ciphertext pairs.}$$

$$X = (x_{ij})_{m \times m}, \quad Y = (y_{ij})_{m \times m}$$

$1 \le j \le m$

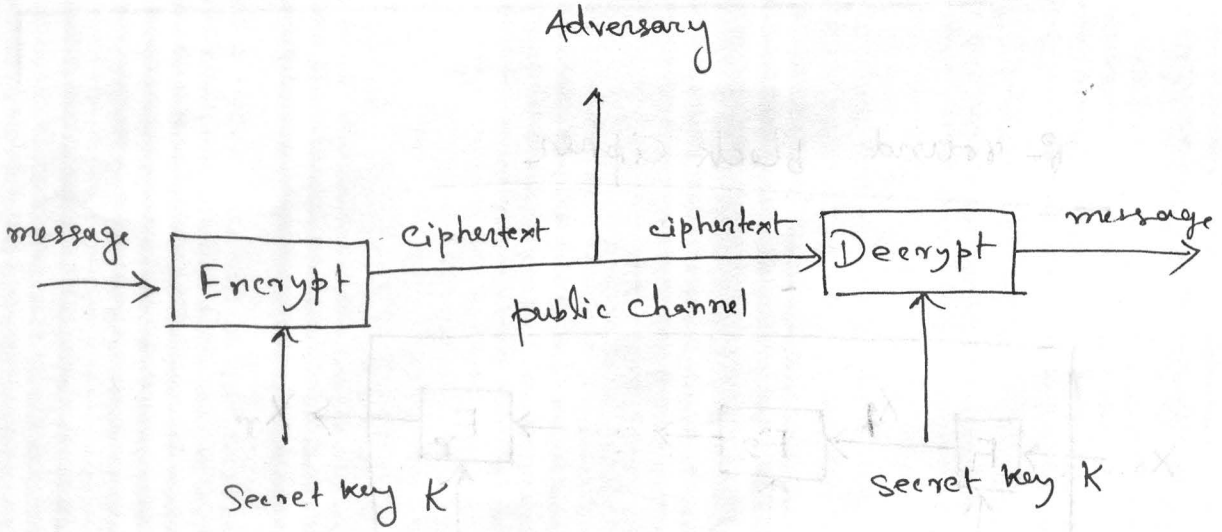
$y_j = e_K(x_j)$

Then $Y = XK$

$\Rightarrow K = X^{-1}Y$ provided X is invertible

- If X is not invertible, then try other sets of m plaintext-ciphertext pairs.

Symmetric Key Encryption



- Block ciphers and Stream ciphers are two types of Symmetric key cryptosystems.

- Block cipher

  - plaintext string $x = x_1 x_2 \cdots$
  - Successive plaintext elements are encrypted using the same key K:

    Key : K

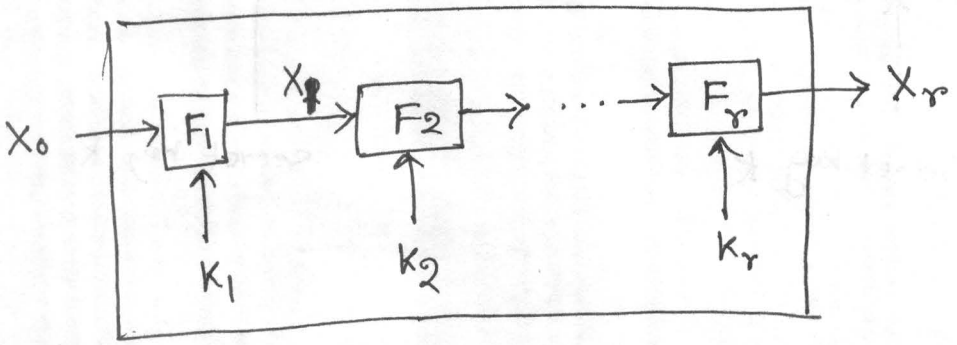
    plaintext : x_1 x_2 x_3 ....

    Ciphertext : $e_K(x_1)$ $e_K(x_2)$ $e_K(x_3)$ ....

  - ciphertext string $y = y_1 y_2 y_3 \cdots = e_K(x_1) e_K(x_2) \cdots$

- Examples : SPN, DES, Rijndael (The AES), IDEA, RC6, and many more ...

r- round block cipher



- $K_1, K_2, \ldots, K_r$ → round keys derived from the secret key using a publicly known key scheduling algm.

- $X_i = F_i(X_{i-1}, K_i)$

- Encryption:
  - X → a plaintext
  - F → round fun. ∀ the rounds
  - encryption operations:

$$X_0 \leftarrow X$$
$$X_1 \leftarrow F(X_0, K_1)$$
$$X_2 \leftarrow F(X_1, K_2)$$
$$\vdots$$
$$X_r \leftarrow F(X_{r-1}, K_r)$$

  - Ciphertext $Y = X_r$

- Decryption
  - F → injective if its second argument is fixed.
  - ∃ a fun. F^{-1} s.t. $F^{-1}(F(X,Y), Y) = X$.
  - Decryption operations:

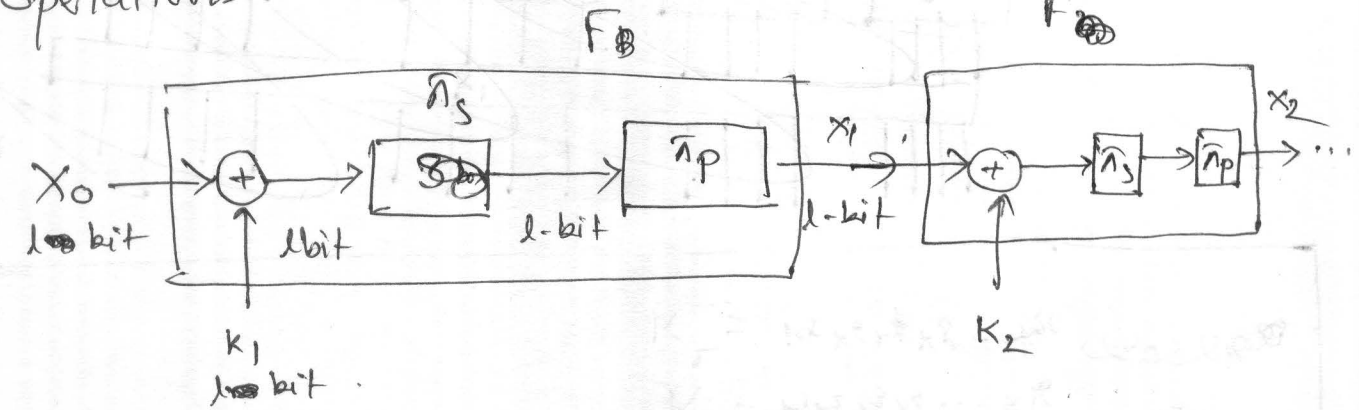
$$X_r \leftarrow Y$$
$$X_{r-1} \leftarrow F^{-1}(X_r, K_r)$$
$$\vdots$$
$$X_1 \leftarrow F^{-1}(X_2, K_2)$$
$$X_0 \leftarrow F^{-1}(X_1, K_1)$$
$$X \leftarrow X_0$$

Iterated cipher

- needs specification of round fun. & a key schedule.
- encryption of a plaintext will proceed using r similar rounds.

SPN (Substitution - Permutation Network)

- uses a sequence of permutation & substitution operations.



$\pi_S \rightarrow$ S box $\rightarrow$ Substitution box.

$\pi_P \rightarrow$ permutation.

- Different S boxes may be used for different rounds.
  $\rightarrow$ different F^0 functions
- DES $\rightarrow$ uses different S boxes for different rounds (Data Encryption Standard)
- AES $\rightarrow$ uses single S box in each rounds & in addition to permutation operations, uses invertible linear transform in each round. (Advanced Encryption Standard).

SPN $\pi_S : \{0,1\}^l \to \{0,1\}^l$.

$\pi_P : \{0,1\}^{lm} \to \{0,1\}^{lm}$.

$\mathcal{P} = \mathcal{C} = \{0,1\}^{lm}$, $\mathcal{K} \subseteq \left(\{0,1\}^{lm}\right)^{r+1}$

initial key K (key) $\xrightarrow{\text{Scheduling algm.}}$ round keys $K^1, \ldots, K^{r+1} \in \{0,1\}^{lm}$.

e.g. $K \in \{0,1\}^{32}$, $K^1, \ldots, K^5 \in \{0,1\}^{16}$.

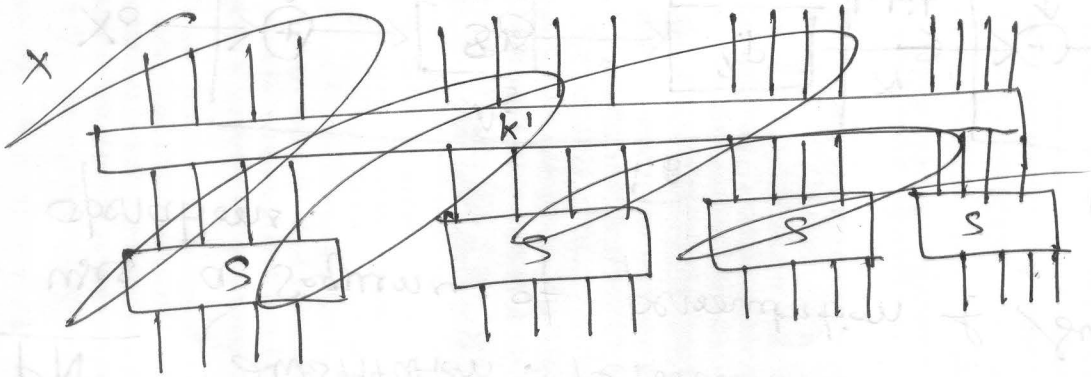
$l = m = r = 4$

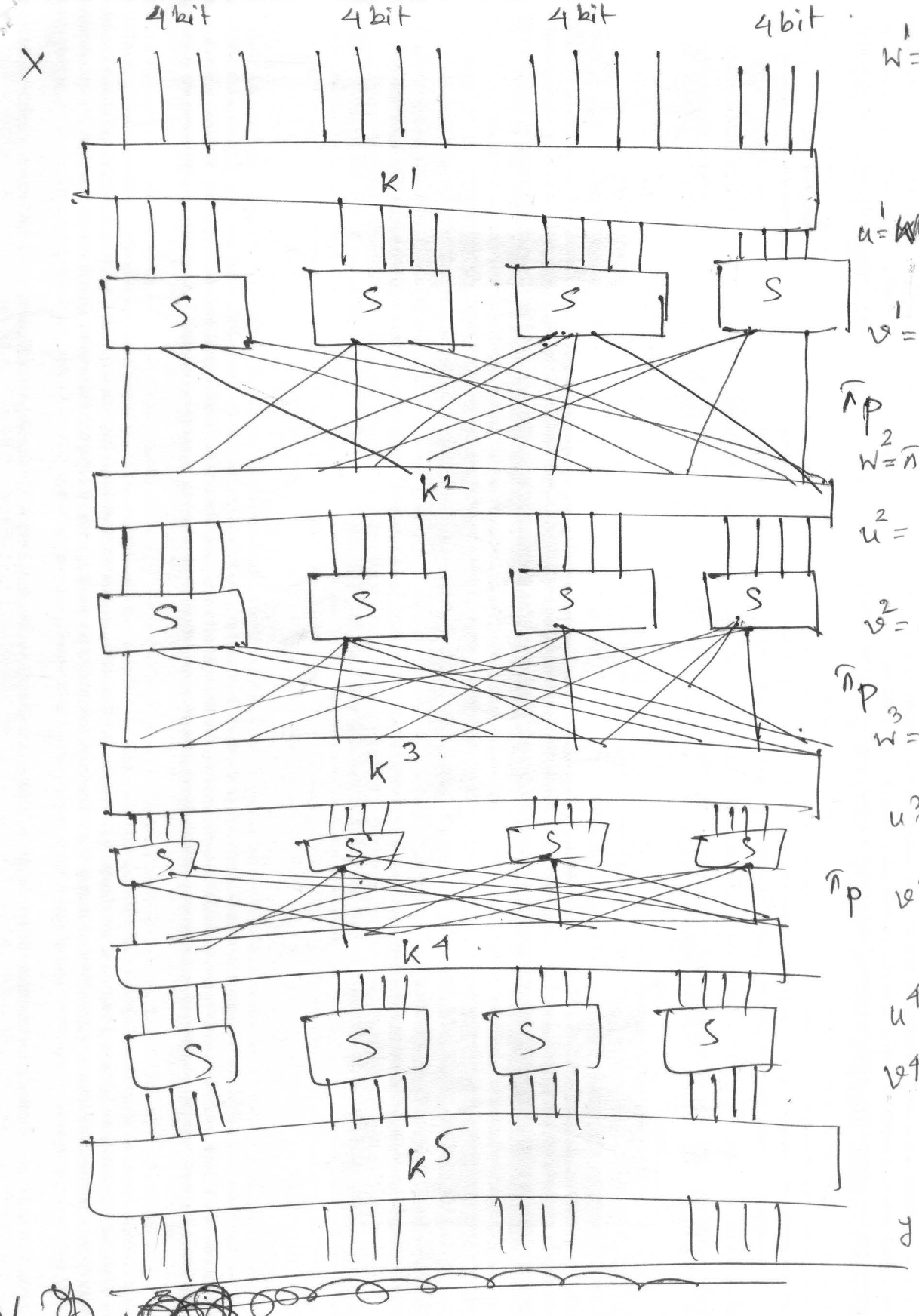
$K = k_1 k_2 k_3 k_4 k_5 \cdots k_{32}$

K^i = 16 consecutive bits of K from k_{4i-3}, $1 \le i \le 5$.

$K^1 = k_1 k_2 k_3 k_4 \cdots k_{16}$

$K^2 = k_5 k_6 k_7 k_8 \cdots k_{20}$



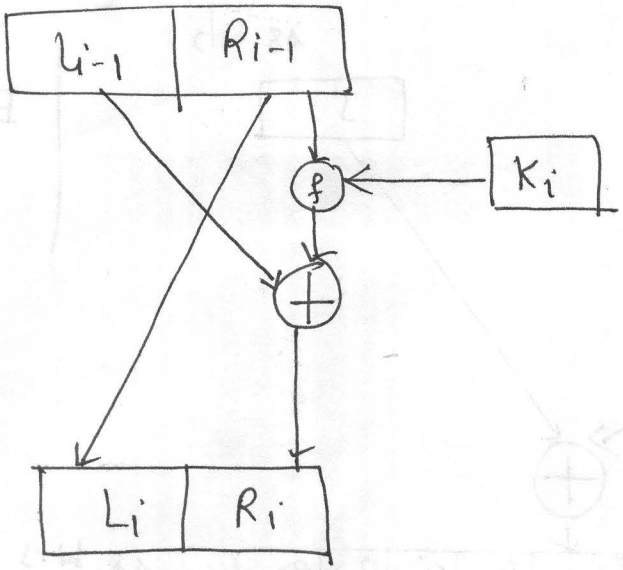


z :	0	1	2	3	...	16
$\pi_P(z)$:	1	5	9		...	16

z :	0	1	2 ... 9	A ... F
$\pi_S(z)$:	E	4	D .. A	6 .. 7

Feistel cipher

plaintext → 64 bits
key → 56 bits
ciphertext → 64 bits.



One round of DES encryption

1. plaintext → x , IP → initial permutation

$x_0 = IP(x) = L_0 R_0$

x_0: 64 bits; L_0: 32 bits; R_0: 32 bits

2. 16 iterations/rounds

for $1 \le i \le 16$

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

R_{i-1}: 32 bits; K_i: 48 bits; f: 32 bits

$K_1, K_2, \ldots, K_{16}$ → each 48 bits.

Computed as a function of the secret key K by key scheduling algm.

K_i → a permuted selection of bits from K.

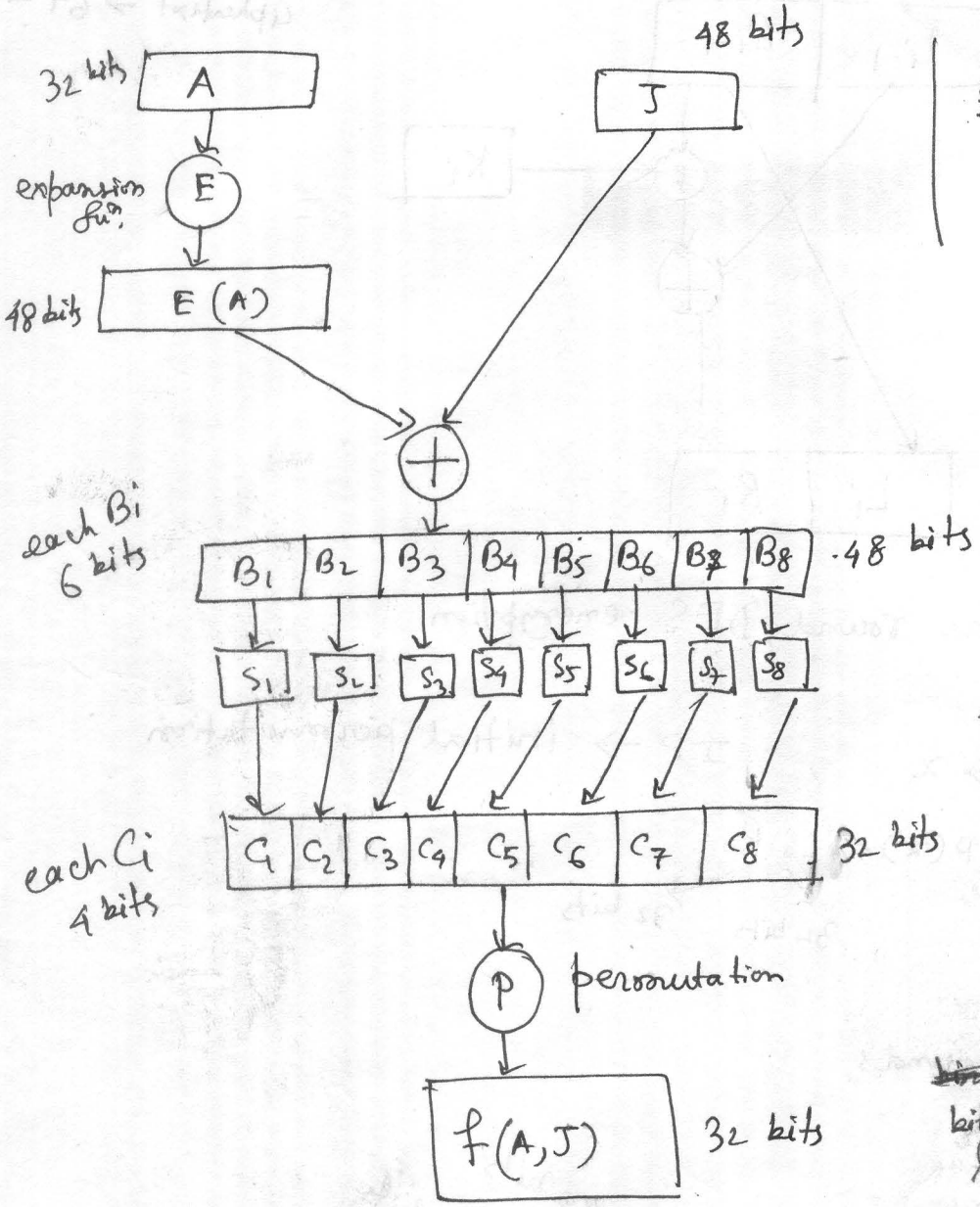
3. $y = IP^{-1}(R_{16} L_{16})$.

Note. ~~Feistel~~ Feistel-type round fun. is always invertible, given the round key:

$L_{i-1} = R_i \oplus f(R_{i-1}, K_i) = R_i \oplus f(L_i, K_i)$, $R_{i-1} = L_i$

$f(A, J) \rightarrow$ 32 bits

(A: 32 bits, J: 48 bits)



$E(A) \rightarrow$ permutated A with 16 bits appearing twice.

$S_i \rightarrow 4 \times 16$ array

each row entries 0–15 of

$B_j = b_1 b_2 b_3 b_4 b_5 b_6$

$S_j(B_j) = S_j(r, c)$

$r \rightarrow$ binary representation of $b_1 b_6$

bit string of length 4.

$c \rightarrow$ binary representation of $b_2 b_3 b_4 b_5$

$0 \le r \le 3$

$0 \le c \le 15$

The DES f function

if $C = (C_1, C_2, \ldots, C_{32})$

then $P(C) = C_{16} C_7 C_{20} C_{21} C_{29} C_{12} \ldots C_{11} C_4 C_{25})$

- P, E, $S_1, \ldots, S_8$

$$P = \begin{bmatrix} 16 & 7 & 20 & 21 \\ 29 & 12 & 28 & 17 \\ 1 & 15 & 23 & 26 \\ 5 & 18 & 31 & 26 \\ 2 & 8 & 24 & 14 \\ 32 & 27 & 3 & 9 \\ 19 & 13 & 30 & 6 \\ 22 & 11 & 4 & 25 \end{bmatrix}$$

$$E = \begin{bmatrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{bmatrix}$$

(11)

S_1 (4×16)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2, S_3, S_4, S_5, S_6

→ substitution boxes.

$b_1 b_2 b_3 b_4 b_5 b_6$

bin

$b_1 b_6 \rightarrow 2$

$b_2 b_3 b_4 b_5 \rightarrow 4$

$(2,4) \rightarrow 13 \rightarrow$ bin.

C_1

- **Stream Cipher**

  - plaintext string $x = x_1 x_2 \cdots$
  - generate a keystream $k_1 k_2 k_3 \cdots$ from the key K:

Key:	K			
keystream:	k_1	k_2	k_3	$\cdots$
plaintext:	x_1	x_2	x_3	$\cdots$
ciphertext:	$e_{k_1}(x_1)$	$e_{k_2}(x_2)$	$e_{k_3}(x_3)$	$\cdots$

  - Ciphertext string $y = y_1 y_2 y_3 \cdots = e_{k_1}(x_1) e_{k_2}(x_2) e_{k_3}(x_3) \cdots$

Remark → Block cipher is a special case of stream ciphers, where the keystream is constant. i.e. $k_i = K \; \forall \; i \geq 1$.

- Example:

  - Consider a message to be a bit stream $m_1 m_2 \cdots$
  - Let $k_1, k_2, \cdots$ be a sequence of pseudorandom bits.
  - Encrypt: $c_i = m_i \oplus k_i$
  - The cipher is: $c_1 c_2 \cdots$
  - Decrypt: $m_i = c_i \oplus k_i$

P: 1000110101011111011
K: 0100101011010011101
C: 1100011110001101110

- Security depends upon the sequence ~~of keys~~ $k_1, k_2, \cdots$
- If k_i is a true random sequence, then the cipher is called an one-time pad.
- One-time pad possesses perfect secrecy

(i.e. $P(x|y) = P(x) \quad \forall \; x \in \mathcal{P} \; \& \; y \in \mathcal{C}$.)

Example (Stream cipher)

(Vigenere Cipher)

$P = C = \mathcal{L} = Z_{26}$, $\mathcal{K} = (Z_{26})^m$.

↓

keystream alphabet.

if $K = (\alpha_1, \alpha_2, \cdots, \alpha_m)$, then define the keystream

$$K_i = \begin{cases} \alpha_i, & 1 \le i \le m \\ K_{i-m}, & i \ge m+1 \end{cases}$$

Generated keystream

$K_1 K_2 \cdots K_m \; K_1 K_2 \cdots K_m \; K_1 K_2 \cdots$

plaintext $x_1 x_2 x_3 \cdots$

keystream $K_1 K_2 K_3 \cdots$

ciphertext $e_{K_i}(x_i) = (x_i + K_i) \bmod 26.$

deciphering $d_{K_i}(y_i) = (y_i - K_i) \bmod 26.$

from the key $K = (\alpha_1, \alpha_2, \cdots, \alpha_m)$.

Periodic Stream cipher with period d

$K_{i+d} = K_d \quad \forall \; i \ge 1.$

period d.

- Vigenere cipher of keyword length m
  - → period m
  - → keystream having short period, can be cryptanalyzed.
- keystream with large period is desirable.

Example: Binary alphabet.

Another method of generating synchronous keystream

Use linear recurrence of degree m

$K = (\alpha_1, \cdots, \alpha_m)$ ⟶ key

keystream $K_i = \alpha_i, \; 1 \le i \le m$

$$K_{i+m} = \sum_{j=0}^{m-1} C_j K_{i+j} \bmod 2$$

$C_0 = 1$

$\forall i \ge 1, \; C_0, C_1, \cdots, C_{m-1} \in Z_2.$

Keystream with long period (generation from a short key).

Key $\rightarrow$ $K = \{(\alpha_1, \cdots, \alpha_m), (c_0, c_1, \cdots, c_{m-1})\}$ $\rightarrow$ $2m$ values
($(\alpha_1, \cdots, \alpha_m)$ $\rightarrow$ IV)

Keystream $\rightarrow$ $k_1, k_2, \cdots$

$k_i = \alpha_i, \quad 1 \le i \le m$.

$\forall i \ge 1$, $\boxed{k_{i+m} = \sum_{j=0}^{m-1} c_j k_{i+j}}$ $\rightarrow$ linear recurrence of degree m

$c_0, c_1, \cdots, c_{m-1} \in Z_2$

$\rightarrow$ each term depends on the previous m terms.

- $c_0 \neq 0$ $\rightarrow$ k_{i+m} is a linear fn of previous terms.
- $(\alpha_1, \cdots, \alpha_m) \neq (0, 0, \cdots 0)$ (o.w. keystream consists of only 0's)
- $c_0, c_1, \cdots, c_{m-1}$ may be chosen in a suitable way to give rise to a periodic keystream with period $2^m - 1$.
- Thus short key can give rise to a keystream having a very long period (desirable).

Illustration $m = 4$.

$k_{i+4} = (k_i + k_{i+1}) \bmod 2, \quad i \ge 1$.

IV $\rightarrow$ any 4-tuple other than $(0, 0, 0, 0)$

Let $IV = (1, 0, 0, 0) = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$

$k_5 = k_1 + k_2$

1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 ...

Keystream

k_1 k_2 k_3 k_4

1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 ...

(15)

• Hardware implementation of keystream

Using LFSR (linear feedback shift register).

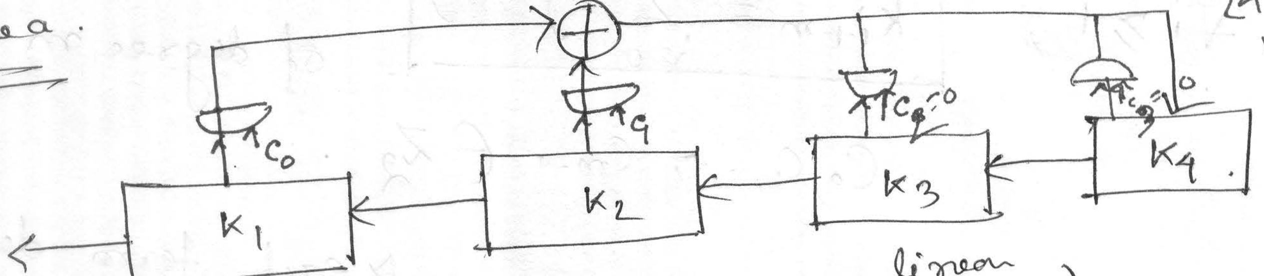
(To be discussed later in detail)

Connection poly → $c(x) = 1 + c_3 x + c_2 x^2 + c_1 x^3 + c_0 x^4$ (where $c_3 = 0$, $c_2 = 0$, $c_1 = 1$, $c_0 = 1$)

$= 1 + c_1 x^3 + c_0 x^4$

$\langle 4, c(x) \rangle$ LFSR

Idea:

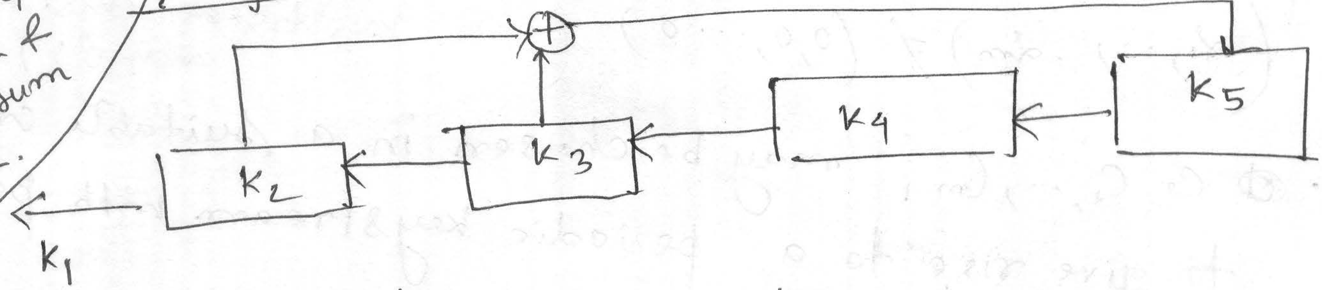


$k_5 = c_0 k_1 + c_1 k_2 + c_2 k_3 + c_3 k_4$ (linear feedback)

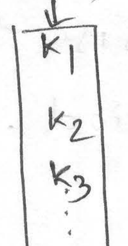
(with $c_0 = 1$, $c_1 = 1$, $c_2 = 0$, $c_3 = 0$)

Carried out by ~~generated by~~ tapping certain stages of the register & taking sum mod 2.

Shift left



Output (Keystream)



k_1	k_2	k_3	k_4
k_2	k_3	k_4	k_5
k_3	k_4	k_5	k_6
k_4	k_5	k_6	k_7

$IV = (k_1, k_2, k_3, k_4) \neq (0,0,0,0)$

Keystream of period $= 2^4 - 1 = 15$ at most.

Max^m. possible period with m-bit LFSR → $2^m - 1$.

→ depends on the choice of ~~connection polynomial~~ $c_0, c_1, \ldots, c_{m-1}$.

$$k_{i+m} = \sum_{j=0}^{m-1} c_j k_{i+j} \pmod 2, \quad i \geq 1$$

Linear Feedback Shift Registers (LFSR).

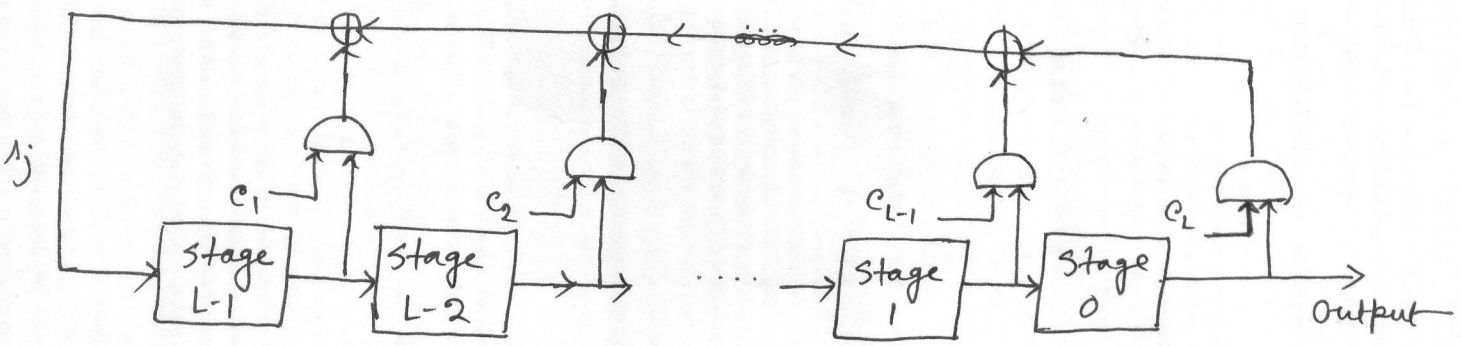
Def^n: A linear feedback shift register (LFSR) of length L consists of L stages numbered $0, 1, \ldots, L-1$, each capable of storing one bit and having one input & one output; and a clock which controls the movement of data.

During each unit of time the following operations are performed:

(i) the content of stage 0 is output & forms part of the output sequence;

(ii) the content of stage i is moved to stage $i-1$ for each i, $1 \le i \le L-1$; and

(iii) the new content of stage $L-1$ is the feedback bit s_j which is calculated by adding together modulo 2 the previous contents of a fixed subset of stages $0, 1, \ldots, L-1$.



A linear feedback shift register (LFSR) of length L

- each c_i is either 0 or 1
- D → AND gate.
- s_j → feedback bit, the modulo 2 sum of the contents of those stages i, $0 \le i \le L-1$, for which $c_{L-i} = 1$.

Def^n: The LFSR in the above figure is denoted by $\langle L, C(D) \rangle$, where

$$C(D) = 1 + C_1 D + C_2 D^2 + \cdots + C_L D^L \in Z_2[D]$$

is the connection polynomial.

The LFSR is said to be non-singular if the degree of $C(D)$ is L (i.e. $C_L = 1$).

If the initial content of stage i is $s_i \in \{0,1\}$ for each i, $0 \le i \le L-1$, then $[s_{L-1}, \ldots, s_1, s_0]$ is called the initial state of the LFSR.

Fact If the initial state of the LFSR in the figure is $[s_{L-1}, \ldots, s_1, s_0]$, then the output sequence (the keystream)

$$s = s_0, s_1, s_2, \ldots$$

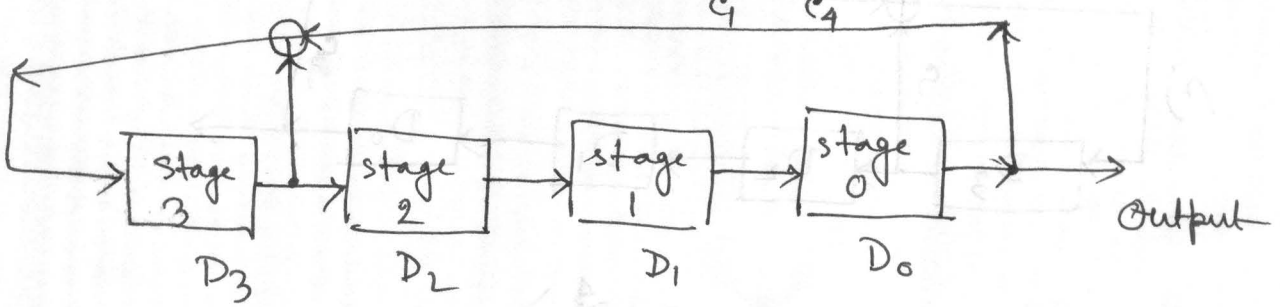
is uniquely determined by the following recursion:

$$s_j = (C_1 s_{j-1} + C_2 s_{j-2} + \cdots + C_L s_{j-L}) \bmod 2$$

for $j \ge L$.

Example

- Consider an LFSR $\langle 4, 1 + D + D^4 \rangle$



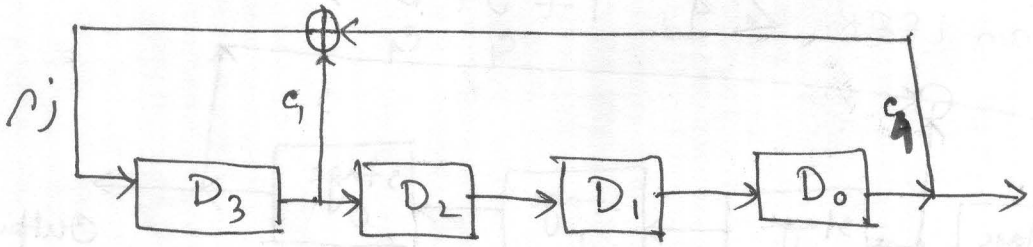
- Induced recurrence: $s_j = s_{j-1} + s_{j-4}$
- If the initial state of the LFSR is $[0,0,0,0]$, then the output sequence is the zero sequence.

- If the keystream is initialized with any vector other than $[0,0,0,0]$, then we obtain a keystream of period 15.
- for example, starting with the initial state $[0,1,1,0]$, the content of stages D_3, D_2, D_1, D_0 at the end of each unit of time t are shown below.

t	D_3 ✓	D_2	D_1	D_0 ✓ → part of the output sequence
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
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
16	0	0	1	1

Repeating.

Output sequence / keystream



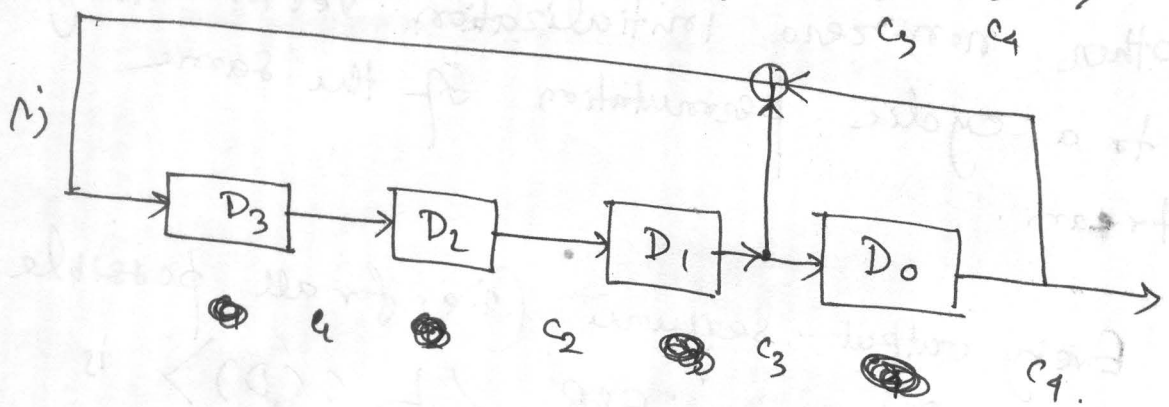
LFSR $\langle 4, 1+D+D^4 \rangle$ (c_1, c_4)

$$s_j = s_{j-1} + s_{j-4}$$

Output: 011001000111101 → satisfies Golomb's randomness postulate. f is periodic with period 15.

Example

LFSR $\langle 4, 1+D^3+D^4 \rangle$



$s_j = s_{j-3} + s_{j-4}$

t	D_3	D_2	D_1 ✓	D_0 ✓
0	1	0	0	1
1	0	0	0	0
2	0	1	0	0
3	0	0	1	0
4	1	0	0	1
5	1	1	0	0
6	0	1	1	0
7	1	0	1	1
8	0	1	0	1
9	1	0	1	0
10	1	1	0	1
11	1	1	1	0
12	1	1	1	1
13	0	1	1	1
14	0	0	1	1
Repeating 15	0	0	0	1
16	1	0	0	0

↓ keystream

Output : 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 ....

- Any other non-zero initialization vector will give rise to a cyclic permutation of the same keystream.

- **Fact**: Every output sequence (i.e. for all possible initial states) of an $LFSR_\infty \langle L, c(D) \rangle$ is periodic iff the connection polynomial $c(D)$ has degree L.

- **fact** N = period of an LFSR $\langle L, c(D) \rangle$.
  - $\langle L, c(D) \rangle$ → degree L
  - N → not always $2^L - 1$.

$c(D)$ irreducible → $N \mid 2^L - 1$.

$c(D)$ primitive poly. of degree L → $N = 2^L - 1$ → maximal length LFSR.

$F_{2^L} = Z_2[D] / (c(D))$.

D is a generator of $F_{2^L}^*$.

maximal length LFSR → the output sequence is called m-sequence → having nice statistical properties.

• Fact (periods of LFSR output sequences).
Let $c(D) \in Z_2[D]$ be a connection polynomial of degree L

(i) If $c(D)$ is irreducible over Z_2, then each of the 2^L-1 non-zero initial states of the non-singular LFSR $\langle L, c(D) \rangle$ produces an output sequence with period equal to the least positive integer N such that $c(D)$ divides D^N+1 in $Z_2[D]$.

(Note: it is always the case that this N is a divisor of 2^L-1).

$N \mid 2^L-1$ (N: period)
$c(D) \mid D^N+1$

(ii) If $c(D)$ is a primitive polynomial, then each of the 2^L-1 non-zero initial states of the non-singular LFSR $\langle L, c(D) \rangle$ produces an output sequence with maximum possible period 2^L-1.

• Defn: If $c(D) \in Z_2[D]$ is a primitive polynomial of degree L, then $\langle L, c(D) \rangle$ is called a maximal length LFSR. The output of a maximal-length LFSR with non-zero initial state is called an m-sequence.

• fact (Statistical properties of m-sequences): Let S be an m-sequence that is generated by a maximum-length LFSR of length L.

(i) S satisfies Golomb's randomness postulates.

(ii) Let k be an integer, $1 \le k \le L$, and let $\bar{s}$ be any subsequence of length 2^k+k-2. Then each non-zero sequence of length k

- Primitive polynomials

Fact Let p be a prime & let the distinct prime factors of $p^m - 1$ be $r_1, r_2, \ldots, r_t$. Then an irreducible polynomial $f(x) \in Z_p[x]$ is primitive iff for each i, $1 \le i \le t$, (↳ deg $f(x) = m$.)

$$x^{(p^m-1)/r_i} \neq 1 \pmod{f(x)}.$$

i.e. x is an element of order $p^m - 1$ in the field

$$Z_p[x] / (f(x)).$$

↳ $0, \alpha, \alpha^2, \ldots \alpha^{m-1}$, $o(\alpha) = p^m - 1$

$g(x) = a_0 + a_1 x + \cdots + a_{m-1} x^{m-1}$
such $g(x) \to p^m$.

---

appears exactly 2^{L-k} times as a subsequence of $\overline{s}$.
Furthermore, the zero sequence of length k appears exactly $2^{L-k} - 1$ times as a subsequence of $\overline{s}$.
In other words, the distribution of patterns having fixed length of at most L is almost uniform.

- Key Agreement; DH, Joux's protocol.
- Secret sharing.

- One-time pads are impractical as:
  - n bits of key is required to encrypt n bits of plaintext (~~key size is big~~)
  - each key is used for only one encryption (this is why the name 'one-time')
  - vulnerable to a known-plaintext attack, since K can be computed as the XOR of the bitstring x & $e_K(x)$. Hence a new key needs to be generated & communicated over a secure channel for every message that is going to be sent. — severe key management prob.

Vernam One-time Pad

- $n \geq 1$
- $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$
- if $x = (x_1, \cdots, x_n)$ & $K = (K_1, K_2, \cdots, K_n)$, then

$$e_K(x) = (x_1 \oplus K_1, x_2 \oplus K_2, \cdots, x_n \oplus K_n) \bmod 2$$

- Decryption is identical to encryption. If $y = (y_1, \cdots, y_n)$, then

$$d_K(y) = (y_1 \oplus K_1, y_2 \oplus K_2, \cdots, y_n \oplus K_n) \bmod 2.$$

- The One-time pad is unconditionally secure

⇓

the cryptosystem cannot be broken even with infinite computational resources.

- Security of unconditionally secure cryptosystems depends on the fact that each key is used for only one encryption.
- Theorem (Shannon's ~~notion of Perfect Secrecy~~)

Suppose $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is a cryptosystem where $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. Then the cryptosystem provides perfect secrecy iff every key is used with equal probability $\frac{1}{|\mathcal{K}|}$, and for every $x \in \mathcal{P}$ & $y \in \mathcal{P}$, there is unique key K s.t. $e_K(x) = y$.

Shannon's Notion of Perfect Secrecy

- $Pr(x|y) = Pr(x)$ for all x & for all y where x is a plaintext & y is a ciphertext.
- The basic strength of stream-cipher lies in how "random" the key stream is.
- If k_i is a true random sequence, then the cipher is called an one-time pad
- One-time pad possesses perfect secrecy. However, one-time pad is impractical.
- Main objective of a stream cipher construction is to get k_i as random as possible.
- Thus random numbers play an important role in the area of stream cipher
- Two criteria are used to validate the randomness of a sequence of numbers:
  - (i) Uniform distribution - the frequency of occurrence of each of the numbers should be approximately the same.
  - (ii) Independence - no one value in the sequence can be inferred from the others.
- Many tests to determine whether a sequence has uniform distribution. No test to prove independence
- Number of tests can be applied to demonstrate that a sequence does not exhibit independence.
- Sources of true random numbers are hard to come by.

- Deterministic ~~algm~~ algorithmic techniques are used instead and if the algm. is good, the resulting sequence will pass many tests of randomness.
- These are called Pseudo Random Number Generators (PRNG).

Example - LFSR.

Hardware implementation → simplicity
→ speed.

→ generated sequence exhibits good statistical properties.
→ m sequence, primitive poly.

Example - NFSR (non-linear feedback shift register)

Hardware oriented Stream ciphers → Grain (uses LFSR, NFSR)
Bivium, Trivium, Mickey

→ candidates of eSTREAM project.
(launched in 2004)
completed in 2008

→ winner

eSTREAR Portfolio of new Stream ciphers

~~Portfolio~~ profile 1 (SW)	Profile 2 (HW)
HC-128	Grain V1
Rabbit	MICKEY 2.0
Salsa 20/12	Trivium
SOSEMANUK	

fault attack.

- Attacks - algebraic attacks (Gröbner basis, SAT Solver)

(Perfect Secrecy)

- One bit encryption

$$C = P \oplus K$$

$$Pr(K=0) = Pr(K=1) = \frac{1}{2}$$

let $Pr(P=0) = 0.6$ & $Pr(P=1) = 0.4$

Then $Pr(P=0 \mid C=1) = \dfrac{Pr(P=0, C=1)}{Pr(C=1)}$

$$= \frac{Pr(P=0, C=1)}{Pr(P=0, C=1) + Pr(P=1, C=1)}$$

$$= \frac{Pr(C=1 \mid P=0)\, Pr(P=0)}{Pr(P=0, C=1) + Pr(P=1, C=1)}$$

$$= \frac{Pr(K=1)\, Pr(P=0)}{}$$

P 0
K
C 1

$$\frac{Pr(C=1 \mid P=0)\, Pr(P=0)}{Pr(C=1 \mid P=0)\, Pr(P=0) + Pr(C=1 \mid P=1)\, Pr(P=1)}$$

$Pr(C=0|P=0)Pr(P=0)$ = $Pr(C=0|P=0)Pr(P=0) + Pr(C=0|P=1)Pr(P=1)$

$$\frac{Pr(K=1)\, Pr(P=0)}{Pr(K=1)\, Pr(P=0) + Pr(K=0)\, Pr(P=1)}$$

$Pr(K=0)Pr(P=0)$ = $Pr(K=0)Pr(P=0) + Pr(K=1)Pr(P=1)$

$$= \frac{\frac{1}{2} \times 0.6}{\frac{1}{2} \times 0.6 + \frac{1}{2} \times 0.4} = 0.6 = Pr(P=0).$$

$\frac{.6}{.6+.4} = .6$

$(P=0|C=0) \overset{?}{=} Pr(P=0)$

$(P=1|C=1) \overset{?}{=} Pr(P=1)$

$(P=1|C=0) \overset{?}{=} Pr(P=1)$.

$$= \frac{Pr(C=1|P=1)\, Pr(P=1)}{Pr(C=1|P=0)\, Pr(P=0) + Pr(C=1|P=1)\, Pr(P=1)}$$

$$= \frac{Pr(K=0)\, Pr(P=1)}{Pr(K=1)\, Pr(P=0) + Pr(K=0)\, Pr(P=1)} = \frac{.4}{.6+.4} = .4$$

Randomness Measurements

- Randomness of sequence : unpredictable property of sequence
- Aim is to measure randomness of the sequence generated by a deterministic method called a generator.
- The test is performed by taking a sample output sequence and subjecting it to various statistical tests to determine whether the sequence possess certain kinds of attributes, a truly random sequence would be likely to exhibit.
- This is the reason the sequence is called pseudo-random sequence instead of a random sequence and the generator is called pseudo-random sequence generator (PRSG).
- The sequence $S = s_0, s_1, s_2, \ldots$ is said to be periodic if there is some positive integer N s.t. $s_{i+N} = s_i$ and smallest N is called the period of sequence.
- Golomb's Randomness Postulates is one of the initial attempts to establish some necessary conditions for a periodic pseudorandom sequence to look random.
- Golomb's Randomness Postulates

R-1 : In every period, the number of 1's differs from the number of 0's by at most 1. Thus

$$\left| \sum_{i=0}^{N-1} (-1)^{s_i} \right| \leq 1.$$

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

R-3: The auto-correlation function

$$c(\tau) = \sum_{i=0}^{N-1} (-1)^{s_i + s_{i+\tau}}$$

→ # of zeros − # of one's in $x - y$ (x: given string, y: τ-shifted string (circular shift))

is two valued. Explicitly,

$$c(\tau) = \begin{cases} N & \text{if } \tau \equiv 0 \pmod N \\ T & \text{if } \tau \not\equiv 0 \pmod N \end{cases}$$

where T is a constant.

Example: Consider the periodic sequence of period 15 with cycle $s^{15} = 011001000111101$

R-1 There are seven 0's and eight 1's

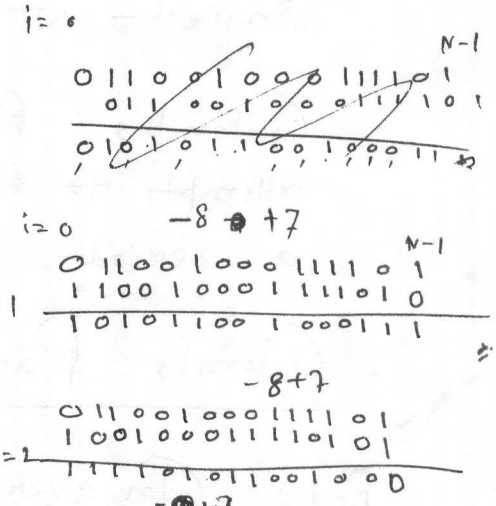
R-2 Total runs is 8

4 runs of length 1

2 runs of length 2

1 run of length 3

1 run of length 4



R-3 The function $c(\tau)$ takes only two values:

$c(0) = 15$ and $c(\tau) = -1$ for $1 \le \tau \le 14$.

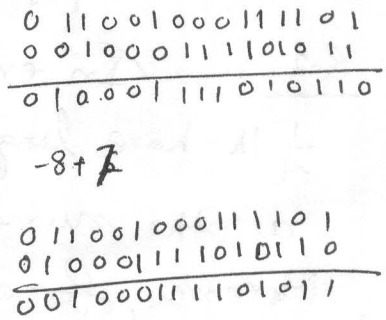
Example: maximal-length LFSR

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

↓

primitive polynomial.

→ generated seq → m sequence.



Golomb's postulates

patterns of length $\le n$ of at most n are fixed uniformly distributed.

Five Basic Tests (Statistical Tests)

- Frequency test (monobit test): To test whether the number of 0's and 1's in sequence s are approximately the same, as would be expected for a random sequence.
- Serial test (two-bit test): To determine whether the number of 00, 01, 10, 11 as subsequences of s are approximately the same, as would be expected for a random sequence.
- Runs test: To determine whether the number of runs of various lengths in the sequence satisfy the R2, as expected for a random sequence.
- Poker test: Let m be a +ve integer. Divide the sequence into $\lfloor \frac{n}{m} \rfloor$ non-overlapping parts of length m. To test whether the number of each sequence of length m are approximately the same, as would be expected for a random sequence.
- Auto-correlation test: To check whether correlation between the sequence and shifted version of it is approximately 0 when the number of shifts is not divisible by the period as expected for a random sequence.

Ref: Handbook of Applied Cryptography. Menezes, ~~Jons~~ Oorschot, Vanstone.

- The statistics used in Frequency Test is

$$X_1 = \frac{(n_0 - n_1)^2}{n}$$

$\sim \chi^2$ distribution with 1 degree of freedom if $n \geq 10$.

n_0 = # of 0's in S
n_1 = # of 1's in S.

- The statistics used in Serial test is

$$X_2 = \frac{4}{n-1}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1$$

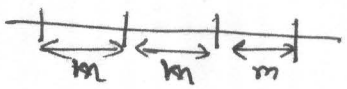
n_{00} = # of occurrences of 00 in S
n_{01} =
n_{10} =
n_{11} =

$\sim \chi^2$ distribution with 2 degrees of freedom if $n \geq 21$ approx.

- Poker test → A statistics is used that follows χ^2 dist. with $2^m - 1$ degrees of freedom.

$$X_3 = \frac{2^m}{K}\sum_{i=1}^{2^m} n_i^2 - K \quad , K = \left[\frac{n}{m}\right]$$

n_i = # of occurrences of ith type of sequence of length m.



$m = 1$ ⇒ Frequency test.

~~χ^2 distribution~~

- Runs test → The statistics used follows χ^2 dist. with some $(2K-2)$ degrees of freedom.
- Autocorrelation test → The statistics used follows an $N(0,1)$ approx. dist.