Pseudo-random Number Generation

DR. ODELU VANGA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRI CITY

Random Numbers in Cryptography

$\boxed{n = pq}$

p, q

- The keystream in the one-time pad
- The secret key in the DES encryption
- The prime numbers p, q in the RSA encryption
- The private key in DSA

DES Encryption:

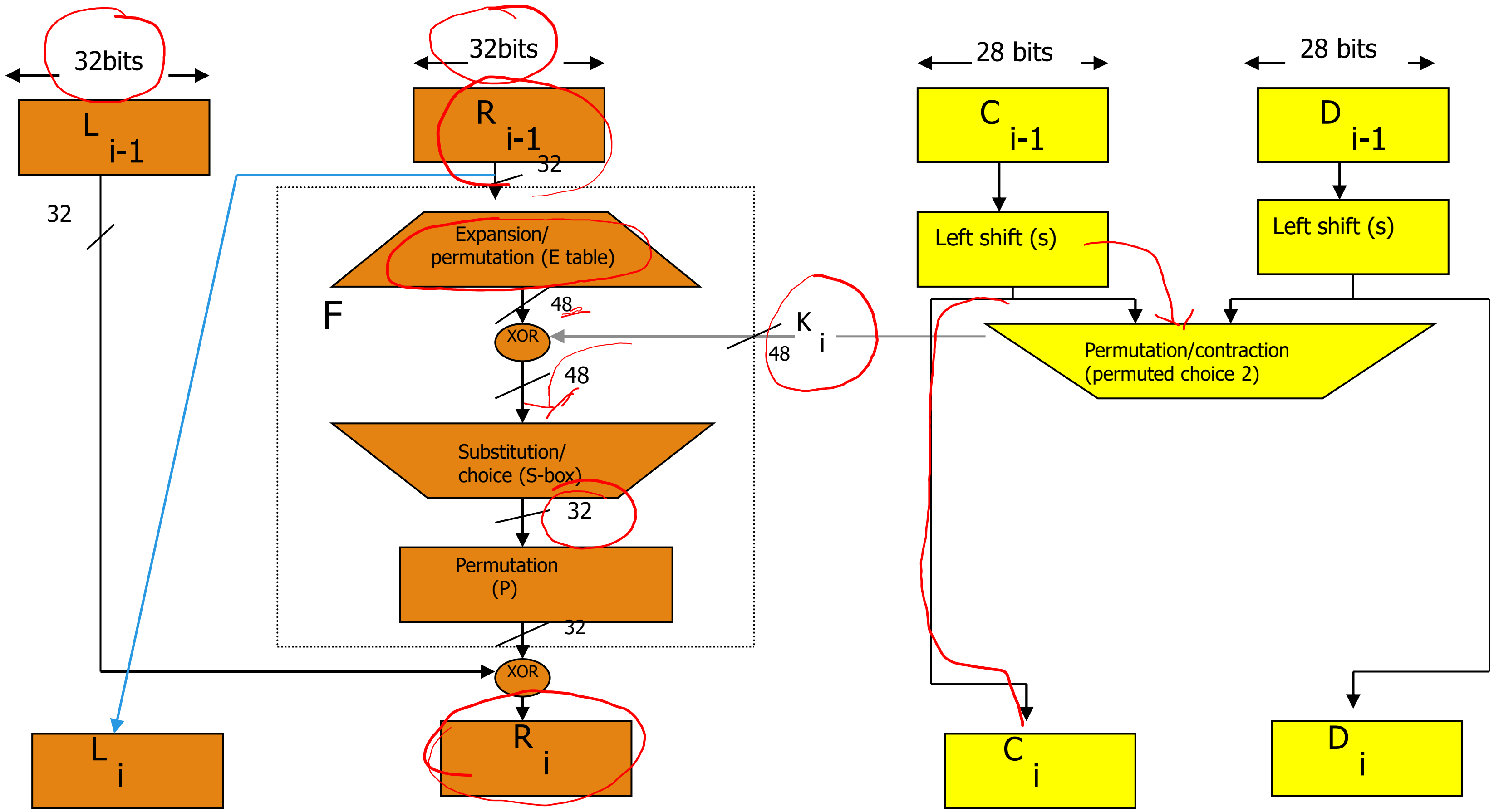
DES: a public standard. But its design criterion has not been published. 64-bit block and 56-bit key

K

64 bit plaintext goes through

- an Initial Permutation (IP).
- 16 Rounds of a complex function f_k as follows:
  - Round 1 of a complex function f_k with sub key K_1 .
  - Round 2 of a complex function f_k with sub key K_2.
  - Round 16 of a complex function f_k with sub key K_{16}
- **At the end of 16 rounds, the Left-half and Right-half are swapped.**
- an Inverse Initial Permutation (IP^{-1}) **to produce 64 bit ciphertext**.

Fig : single Round of DES Algorithm:



i-th Round

The part in yellow, in the previous slide, shows the sub key generation. After PC1, the circular rotations are independent for the left half and the right-half.

ENCRYPTION: In the i-th round,

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, Ki)$$

$$= L_{i-1} \oplus P(S(E(R_{i-1}) \oplus Ki))$$

Where E: expansion from 32 bits to 48

S: Using 8 S-boxes to convert 48 bits to 32 bits – each S box converts 6 bits to 4 bits

P: permutation

Pseudo-random Number Generator

Pseudo-random number generator

- A polynomial-time computable function f(x) that expands
  a ***short random string x*** into
  a ***long string f(x)*** that appears random

Objectives

- Fast
- Secure

Pseudo-random Number Generator

Classical PRNGs

- Linear Congruential Generator

Cryptographically Secure PRNGs

- Blum-Blum-Shub Generator

Linear Congruential Generator - Algorithm

Based on the linear recurrence

$$\boxed{x_i = ax_{i-1} + b \pmod m,\ i \geq 1}$$

$x_1 = a\boxed{x_0} + b$

where $\boxed{x_0 \text{ is the seed}}$ or start value, a is the multiplier

b is the increment, m is the modulus

Output

$(x_1, x_2, \ldots, x_k)$

$\boxed{y_i = x_i \bmod 2}$

$Y = (y_1 y_2 \ldots y_k)$ ← pseudo-random sequence of K bits

$$\left\{\begin{array}{l} x_1 = ax_0 + b \bmod m \\ x_2 = ax_1 + b \bmod m \\ \vdots \\ x_k \end{array}\right.$$

Linear Congruential Generator - Example

Let $\mathbf{x_n = 3x_{n-1} + 5 \bmod 31}$, $n \geq 1$, and $x_0 = 2$

- 3 and 31 are relatively prime, one-to-one (affine cipher)
- 31 is prime, order is 30

$o(2^x_{31}) = 30$

$\phi(n)$

Then we have the 30 residues in a cycle:

- 2, 11, 7, 26, 21, 6, 23, 12, 10, 4, 17, 25, 18, 28, 27, 24, 15, 19, 0, 5, 20, 3, 14, 16, 22, 9, 1, 8, 29, 30

Pseudo-random sequences of 10 bits

- when $x_0 = 2$

  1101010001

- **When $x_0 = 3$**

  0001101001

$x_0 = 3$

$x_1 = 3 \times 3 + 5 \bmod 31$

$= 14 \bmod 31$

$= 14$

$y_1 = x_1 \bmod 2 = 0$

$x_0 = 2.$

$x_1 = 3x_0 + 5 \bmod 31$

$= 3 \times 2 + 5 \bmod 31$

$= 11 \bmod 31$

$= 11$

$y_1 = x_1 \bmod 2 = 1$

Linear Congruential Generator

Fast, but insecure

- Sensitive to the choice of parameters a, b, and m
- **Serial correlation between successive values**

$(a, m) = 1$

$x_1 = a x_0 + b \bmod m$

$x_2 = a x_1 + b \bmod m$

$x_0 = (x_1 - b) a^{-1}$

Applications:

- **Used commonly in compilers: Rand()**
- **Not suitable for high-quality randomness applications**
- **Not suitable for cryptographic applications**

Blum-Blum-Shub Generator - Concept

Quadratic residues

$\sqrt{a}$

- Let p be an odd prime and a be an integer
- a is a quadratic residue modulo p if a is not congruent to 0 mod p and there exists an integer x such that $a \equiv x^2 \bmod p$
- a is a quadratic non-residue modulo p if a is not congruent to 0 mod p and a is not a quadratic residue modulo p

Example

- Let p=5, then $1^2 = 1$, $2^2 = 4$, $3^2 = 4$, $4^2 = 1$
- 1 and 4 are quadratic residues modulo 5
- 2 and 3 are quadratic non-residues modulo 5

Blum-Blum-Shub Generator - Algorithm

Based on the squaring one-way function ⇒ Given x, finding $f(x)$ is easy. but, given $f(x)$, finding x is hard.

- Let p, q be two odd primes and **p≡q≡3 mod 4**
- Let **n = pq, s is a seed.**
- Let $x_0 = s^2 \bmod n$, then define

$$\boxed{x_i = x_{i-1}^2 \bmod n}, \; i \geq 1$$

$p, q \Rightarrow n = pq$

but $n = pq \not\Rightarrow p, q$

$f(x) = x^2 \bmod n$

Output

$(x_1, x_2, \ldots, x_k)$

$y_i = x_i \bmod 2$

$Y = (y_1 y_2 \ldots y_k)$ ← pseudo-random sequence of K bits

Example: p=7, q=11, and n=pq = 77. Let seed s=3.

$x \to x^2$

$x_0 \to x^2$

x_1

Blum-Blum-Shub Generator

Why p≡q≡3 mod 4

- Every quadratic residue x has a square root y which is itself a quadratic residue (y is principal square root of x)
- Denote the square root of x to be y, that is, $x=y^2 \bmod n$
- **Let p= 4m+3, then m=(p-3)/4.**
  - $y = x^{(p+1)/4} \bmod p$ is a principal square root of x modulo p

    $x^{(p-1)/2}=x^{(4m+3-1)/2}=x^{2m+1}=1 \bmod p$ => $\mathbf{x^{2m+2}=x \bmod p}$

    $\Rightarrow (x^{m+1})^2 = x \bmod p \Rightarrow y = x^{m+1} = x^{(p+1)/4}$
  - y is a quadratic residue

    $y^{(p-1)/2}= (x^{(p+1)/4})^{(p-1)/2}= (x^{(p-1)/2})^{(p+1)/4}=1^{(p+1)/4}=1 \bmod p$

unique sol.

- Similar for q, $y = x^{(q+1)/4} \bmod q$
- Since n=pq and x is a quadratic residue modulo n, then x has a unique square root modulo n (**Chinese remainder theorem**)

- As a result, the mapping from **x to x² mod n is a bijection from the set of quadratic residues modulo n** onto itself

Blum-Blum-Shub Generator

Blum-Blum-Shub Generator is provably secure

- Euler's criterion
- Legendre symbol
- Jacobi symbol
- Composite quadratic residues

Blum-Blum-Shub Generator

Euler's criterion

- Let p be an odd prime. Then a is a quadratic residue modulo p if and only if $a^{(p-1)/2} \equiv 1 \bmod p$

Legendre symbol

- Let p be an odd prime and a be an integer

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \pmod{p} \\ 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ -1 & \text{if } a \text{ is a quadratic non-residue modulo } p \end{cases}$$

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}$$

Blum-Blum-Shub Generator

Jacobi symbol

- Let n be an odd positive integer
- **p_i is the prime factor of n** and e_i is the power of the prime factor
- (a/p_i) is the Legendre symbol and (a/n) is the Jacobi symbol

$$\boxed{n = \prod_{i=1}^{k} p_i^{e_i}}$$

$10 = 2 \times 5 = \prod_{i=1}^{2} p_i^{e_i} = 2^1 \times 5^1$

$$\left(\frac{a}{n}\right) = \boxed{\prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}}$$

- Example: Let **n=15=3*5**
  (9/15)=(9/3)(9/5)=0
  (11/15)=(11/3)(11/5)=(2/3)(1/5)=(-1)(1)=-1
  (8/15)=(8/3)(8/5)=(2/3)(3/5)=(-1)(-1)=1

Blum-Blum-Shub Generator

Composite quadratic residues

- Let p, q be two odd primes and **n = pq**
- If (x/n) = (x/p)(x/q) = 1, then
  either (x/p) = (x/q) = 1, x is a quadratic residue modulo n
  or (x/p) = (x/q) = -1, x is a pseudo-square modulo n
- It is difficult to determine whether x is a quadratic residue modulo n, which as difficult as factoring n=pq.

$$\left(\frac{x}{n}\right) = \begin{cases} 0 & \text{if } \gcd(x,n) > 1 \\ 1 & \text{if } \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = 1 \text{ or if } \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1 \\ -1 & \text{if one of } \left(\frac{x}{p}\right) \text{ and } \left(\frac{x}{q}\right) = 1 \text{ and the other} = -1 \end{cases}$$

- Example: Let n=15=3*5
  (8/15)=(8/3)(8/5)=(2/3)(3/5)=(-1)(-1)=1; 8 is a pseudo-square
  (4/15)=(4/3)(4/5)=(1)(1)=1; 4 is a quadratic residue

Thank You