Find multiplicative inverse using Extended Euclidean Algorithm

Condition

$a^{-1} \bmod n$.

$\gcd(a, n)$ should be 1.

For eg $5^{-1} \bmod 10$

$\gcd(5, 10) = 5 \neq 1$.

$\therefore$ Inverse of 5 does not exist under modulo 10.

Ex-1 $3^{-1} \bmod 17$.

Step 1 Find gcd of (3, 17)

$3\overline{)17}(5$
15

$2\overline{)3}(1$
2

1.

$17 = 3 \times 5 + 2$

$3 = 2 \times 1 + 1$

$\therefore \gcd = 1$

$\therefore$ There exists a M.I. for 3 mod 17.

Now, express gcd in terms of $(3x + 17y)$

$1 = 3 - 2 \times 1$

$= 3 - (17 - 3 \times 5) \times 1$

$= 3 - 17 + 3 \times 5$

$= 3 \times (6) + 17 \times (-1)$

$\therefore 3^{-1} \bmod 17 = 6$

3.

a)

i) $17^{-1} \bmod 101$

Step 1: Find gcd of (17, 101)

$$17 \overline{)101} \; 5 \quad (85)$$
$$16 \overline{)17} \; 1 \quad (16)$$
$$1.$$

$\therefore 101 = 17 \times 5 + 16$

$17 = 16 \times 1 + 1$

$\therefore \gcd(17, 101) = 1.$

$\therefore$ M.I. of 17 exists under 101.

Step 2

NOW, lets express gcd in terms of $17x + 101y$.

$$1 = 17 - 16 \times 1$$
$$= 17 - (101 - 17 \times 5) \times 1$$
$$= 17 - 101 + 17 \times 5$$
$$= 17 \times (6) + 101 \times (-1)$$

$\therefore$ $17^{-1} \bmod 101 = 6$ (Ans)

ii) $357^{-1} \bmod 1234$

Step 1 find gcd (357, 1234)

$$
\begin{array}{r|l}
357 & 1234 \\
1071 & 3 \\
\hline
163 & 357 \\
326 & 2 \\
\hline
31 & 163 \\
155 & 5 \\
\hline
8 & 31 \\
24 & 3 \\
\hline
7 & 8 \\
7 & 1 \\
\hline
1. &
\end{array}
$$

$1234 = 357 \times 3 + 163$

$357 = 163 \times 2 + 31$

$163 = 31 \times 5 + 8$

$31 = 8 \times 3 + 7$

$8 = 7 \times 1 + 1$ (gcd)

$\therefore \gcd(357, 1234) = 1$

$\therefore$ M.I. of 357 exists under mod 1234

Step 2 Express gcd in terms of $(357x + 1234y)$

$1 = 8 - 7 \times 1$

$= 8 - (31 - 8 \times 3) \times 1$

$= 8 - 31 + 8 \times 3$

$= 8 \times 4 - 31$

$= (163 - 31 \times 5) \times 4 - 31$

$= 163 \times 4 - 31 \times 20 - 31$

$= 163 \times 4 - 31 \times 21$

$= 163 \times 4 - (357 - 163 \times 2) \times 21$

$= 163 \times 4 - 357 \times 21 + 163 \times 42$

$= -357 \times 21 + 163 \times 46$

$= -357 \times 21 + (1234 - 357 \times 3) \times 46$

$= -357 \times 21 + 1234 \times 46 - 357 \times 138$

$= -357 \times 159 + 1234 \times 46$

$= 357 \times (-159) + 1234 \times 46$

$\therefore 357^{-1} \bmod 1234$

$= -159$

$\equiv (1234 - 159)$

$= 1075$

both can be ans.

$\therefore$ Ans $= -159$ or 1075

iii) $3125^{-1} \bmod 9987$

Step 1 Find gcd of (3125, 9987)

$$\begin{array}{r} 3125 \overline{)9987} (2 \\ 6250 \\ \hline 3737 \end{array} \qquad \begin{array}{r} 3125 \overline{)9987} (3 \\ 9375 \\ \hline 612 \overline{)3125} (5 \\ 3060 \\ \hline 65 \overline{)612} (9 \\ 585 \\ \hline 27 \overline{)65} (2 \\ 54 \\ \hline 11 \overline{)27} (2 \\ 22 \\ \hline 5 \overline{)11} (2 \\ 10 \\ \hline 1 \end{array}$$

$9987 = 3125 \times 3 + 612$

$3125 = 612 \times 5 + 65$

$612 = 65 \times 9 + 27$

$65 = 27 \times 2 + 11$

$27 = 11 \times 2 + 5$

$11 = 5 \times 2 + (1)$

$\therefore \gcd(3125, 9987) = 1$

$\therefore$ M.I. of 3125 exists under modulo 9987

Step 2 Express gcd in terms of $3125x + 9987y$

$1 = 11 - 5 \times 2$

$= 11 - (27 - 11 \times 2) \times 2$

$= 11 - 27 \times 2 + 11 \times 4$

$= 11 \times 5 - 27 \times 2$

$= (65 - 27 \times 2) \times 5 - 27 \times 2$

$= 65 \times 5 - 27 \times 10 - 27 \times 2$

$= 65 \times 5 - 27 \times 12$

$= 65 \times 5 - (612 - 65 \times 9) \times 12$

$= 65 \times 5 - 612 \times 12 + 65 \times 108$

$= 65 \times 113 - 612 \times 12$

$= (3125 - 612 \times 5) \times 113 - 612 \times 12$

$= 3125 \times 113 - 612 \times 565 - 612 \times 12$

$= 3125 \times 113 + 612 \times (-577)$

$= 3125 \times 113 + (9987 - 3125 \times 3) \times (-577)$

$= 3125 \times 113 + 9987 \times (-577) + 3125 \times 1731$

$= 3125 \times 1844 + 9987 \times (-577)$

$\therefore 3125^{-1} \bmod 9987 = 1844$ (Ans)

3

~~3(b)~~ ~~1)~~ 2)

Rules for finding x in linear congruence:

General format: $ax \equiv b \pmod n$

1. Find $\gcd(a, n) = d$ (Euclid)
2. Find $d \bmod n$ → These no. of Soln. are possible
3. b/d → if possible → Solution exist
4. Divide both the sides by d
5. Multiply both sides by 'multiplicative inverse of a' i.e $(a \cdot a^{-1})x = b \cdot a^{-1} \pmod n$
6. General Sol. eqn. is:-

$$x_k = x_0 + k\left(\frac{n}{d}\right)$$

Example

$$14x \equiv 12 \pmod{18}$$

$$ax \equiv b \pmod n$$

$a = 14,\ b = 12,\ n = 18$

1. $\gcd(a, n) \longrightarrow d$

   $\gcd(14, 18) \longrightarrow 2\ (d)$

2. $b/d = 12/2 = 6 \longrightarrow$ Sdn exist

3. $d \bmod n = 2 \% 18 = 2$

   ∴ 2 Soln possible

4. Divide both sides by d.

$$\frac{14x}{2} \equiv \frac{12}{2} \pmod{\frac{18}{2}}$$

$$7x \equiv 6 \pmod 9$$

(5) Multiply by multiplicative inverse of a

$$7x \equiv 6 \pmod 9$$

$$7 \cdot 7^{-1} x \equiv 6 \cdot 7^{-1} \pmod 9$$

$$x \equiv 6 \cdot 7^{-1} \pmod 9$$

Now, we have to find $7^{-1} \bmod 9$.

$\gcd(7,9) = 1$.

$$9 = 7 \times 1 + 2$$

$$7 = 2 \times 3 + 1$$

$$1 = 7 - 2 \times 3$$

$$= 7 - (9 - 7 \times 1) \times 3$$

$$= 7 - 9 \times 3 + 7 \times 3$$

$$= 7 \times 4 + 9 \times (-3)$$

$\therefore 7^{-1} \bmod 9 = 4$.

$$x = 6 \times 4 \pmod 9 = 6$$

$\therefore \boxed{x_0 = 6}$

$$x_K = x_0 + K\left(\frac{n}{d}\right)$$

$$= 6 + K\left(\frac{18}{2}\right)$$

$$= 6 + 9K$$

2 soln pass. $\therefore K = 0, 1$

$\therefore x_0 = 6$

$x_1 = 6 + 9 = 15$

3b) $x \equiv 9 \pmod{26}$

$ax \equiv b \pmod{n}$

$a=1, b=9, n=26$

① $\gcd(a,n) = \gcd(1,26) = 1 = d$

② $b/d = 9/1 = 9$ ∴ Soln exist

③ $d \bmod n = 1 \bmod 26 = 1$. ∴ 1 soln possible

④ Divide both side by d

$$\frac{x}{1} \equiv \frac{9}{1} \pmod{\frac{26}{1}}$$

$$\Rightarrow x \equiv 9 \pmod{26}$$

⑤ ✓

⑥ $x_K = x_0 + K\left(\frac{n}{d}\right)$

$= 9 + K\left(\frac{26}{1}\right)$

Now, only 1 soln for

∴ $K = 0$

∴ $x_0 = 9$.

Similarly $x \equiv 12 \pmod{25}$ $x \equiv 23 \pmod{27}$

$x_0 = 12$ $x_0 = 23$

$15x \equiv 56 \pmod{101}$

$\Rightarrow ax \equiv b \pmod{n}$.

$a = 15,\ b = 56,\ n = 101$

$15)101(6$
$\quad 90$
$\quad 11)15(1$
$\quad\quad 11$
$\quad\quad 4)11(2$
$\quad\quad\quad 8$
$\quad\quad\quad 3)4(1$
$\quad\quad\quad\quad 3$
$\quad\quad\quad\quad 1$

① $\gcd(a,n) = \gcd(15,101) = 1 = d$

② $b/d = 56/1 = 56$ ∴ soln possible

③ $d \bmod n = 1 \bmod 101 = 1$.

∴ 1 soln possible

④ Divide both sides by d

$15x \equiv 56 \pmod{101}$
$\quad T \quad\quad T \quad\quad\quad T$

$\Rightarrow 15x \equiv 56 \pmod{101}$

$15)101(6$
$\quad 90$
$\quad 11)15(1$
$\quad\quad 11$
$\quad\quad 4)11(2$
$\quad\quad\quad 8$
$\quad\quad\quad 3)4(1$
$\quad\quad\quad\quad 3$
$\quad\quad\quad\quad 1$

⑤ $15x \equiv 56 \pmod{101}$

$\Rightarrow 15 \cdot 15^{-1} x \equiv 56 \cdot 15^{-1} \pmod{101}$

$\Rightarrow x \equiv 56 \cdot 15^{-1} \pmod{101}$

Find $15^{-1} \bmod 101$

$\gcd(15,101) = 1$.

$101 = 15 \times 6 + 11$
$15 = 11 \times 1 + 4$
$11 = 4 \times 2 + 3$
$4 = 3 \times 1 + 1$

$1 = 4 - 3 \times 1$
$= 4 - (11 - 4 \times 2) \times 1$
$= 4 - 11 \times 1 + 4 \times 2$
$= 4 \times 3 - 11 \times 1$
$= (15 - 11 \times 1) 3 - 11 \times 1$
$= 15 \times 3 - 11 \times 3 - 11 \times 1$
$= 15 \times 3 - 11 \times 4$
$= 15 \times 3 - (101 - 15 \times 6) \times 4$
$= 15 \times 3 - 101 \times 4 + 15 \times 24$
$= 15 \times 27 + 101(-4)$

∴ $15^{-1} \bmod 101 = 27$

∴ $x_0 \equiv 56 \times 27 \pmod{101}$
$= 98$

Only 1 soln possible

RSA

1. Key Generation

i) Select 2 large prime no.s 'p' and 'q'

ii) Calculate $n = p * q$

iii) calculate $\phi(n) = (p-1) * (q-1)$ // Euler toitient function

iv) Choose value of e $1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$.

v) Calculate $d \equiv e^{-1} \bmod \phi(n)$

i.e $ed \equiv 1 \pmod{\phi(n)}$

vi) public key = $\{e, n\}$

vii) private key = $\{d, n\}$

2 Encryption

$M < n$

$C = M^e \bmod n$

↙ ciphertext ↓ plaintext

3. Decryption

$M = C^d \bmod n$

Note: (e, n) is public key used in encryption.

(d, n) is private key used in decryption.

Example Let $p = 3, q = 11$

$n = p \times q = 3 \times 11 = 33$

$\phi(n) = (p-1)(q-1) = (3-1)(11-1) = 2 \times 10 = 20$

Let, $e = 7$ $\left[\begin{array}{l} 1 < e < \phi(n) \\ \gcd(e, \phi(n)) = 1 \end{array} \right]$

$d = e^{-1} \bmod \phi(n)$

$= 7^{-1} \bmod 20$

Now, let's find $7^{-1} \bmod 20$ by extended Euclid algo.

$\gcd(7, 20) = 1.$

$7)20(2$
14
$6)7(1$
6
1

$20 = 7 \times 2 + 6$

$7 = 6 \times 1 + 1$

$1 = 7 - 6 \times 1$

$= 7 - (20 - 7 \times 2)$

$= 7 - 20 + 7 \times 2$

$= 7 \times 3 + 20 \times (-1)$

$\therefore 7^{-1} \bmod 20 = 3.$

$\therefore d = 3 \bmod 20 = 3$

$\therefore$ Public key $= \{e, n\} = \{7, 33\}$

Private key $= \{d, n\} = \{3, 33\}$ }

Encryption let $M = 31$ $C = M^e \bmod n = (31)^7 \bmod 33 = 4$ ($M < n$)

Decryption $M = C^d \bmod n = (4)^3 \bmod 33 = 31$

Diffie-Hellman Key Exchange

i) Not an encryption algo

ii) used to exchange secret keys b/w 2 users

iii) We will use asymmetric encryption to exchange the secret key b/w users.

Why this algo

b/c when we are sending a key to receiver, it can be attacked in b/w.

Algorithm

i) Consider a prime number 'q'

ii) Select α such that it must be the primitive root of q and $\alpha < q$

'α' is a primitive root of q if

$\alpha \bmod q$
$\alpha^2 \bmod q$ gives res $\{1, 2, 3, \ldots, q-1\}$
$\alpha^{q-1} \bmod q$

iii) Assume X_A (private key of A) and $X_A < q$

calculate $Y_A = \alpha^{X_A} \bmod q$

↓

public key of A

iv) Assume X_B (private key of B) and $X_B < q$

calculate $Y_B = \alpha^{X_B} \bmod q$

↓

public key of B

Now we'll calculate Secret Key

$K_1 = (Y_B)^{X_A} \bmod q$ $\qquad$ $K_2 = (Y_A)^{X_B} \bmod q$

Y_B → public keys ← Y_A

Known to all

If $K_1 = K_2$ then we say key exchange is successful.

So current Scenario

private key

$X_A = 3$

$q = 7$

$\alpha = 5$

public keys

$Y_A = 6$

$Y_B = 2$

private key

$X_B = 4$

$Y_A = \alpha^{X_A} \bmod q = 5^3 \bmod 7 = 3$

$Y_B = \alpha^{X_B} \bmod q = 5^4 \bmod 7 = 2$

$Y_B = (\alpha)^{X} \bmod q$

$3 = (7)^{X} \bmod 11$

$K_1 = (Y_B)^{X_A} \bmod q$

$= (2)^3 \bmod 7$

$= 21$

$K_2 = (Y_A)^{X_B} \bmod q$

$= (6)^4 \bmod 7$

$= 1$

$K_1 = K_2$

EL Gamal Encryption Scheme

q and α

q ↓ prime no.

α ↓ primitive root of q.

User A

1. Select a random integer X_A (private key), $X_A < q$
2. Calculate $Y_A = (\alpha)^{X_A} \bmod q$
3. Private key X_A, public key $= \{q, \alpha, Y_A\}$

User B (encryption)

1. Plaintext M
2. Select a random integer k, $k < q$
3. calculate $K = (Y_A)^k \bmod q$

   K ↓ Secret key; Y_A ↓ public key of user A
4. Calculate $C_1 = (\alpha)^k \bmod q$
5. Calculate $C_2 = KM \bmod q$

ciphertext $= (C_1, C_2)$

User A (Decryption)

ciphertext (C_1, C_2)

1. Calculate $K = (C_1)^{X_A} \bmod q$
2. $M = (C_2 K^{-1}) \bmod q$

Example

$q = 19, \alpha = 10$

User A

$X_A = 5$

$Y_A = (\alpha)^{X_A} \bmod q$

$= (10)^5 \bmod 19$

$= 3$

User B (Encryption)

$M = 17$

Select a random number $k = 6$

$K = (Y_A)^K \bmod q = (3)^6 \bmod 19 = 7$

Calculate $c_1 = (\alpha)^K \bmod q$

$= (10)^6 \bmod 19$

$c_1 = 11$

$c_2 = KM \bmod q$

$= (7 \times 17) \bmod 19$

$= 5$

$c_1 = 11$
$c_2 = 5$

User A (Decryption)

1. $K = (c_1)^{X_A} \bmod q$

$= (11)^5 \bmod 19$

$= 7$

$M = (c_2 K^{-1}) \bmod q$

$= (5 \times 7^{-1}) \bmod 19$

$= (5 \times 11) \bmod 19 = 17$

6. Diffic Hellman

$q = 71, \alpha = 7$

a) $X_A = 69$

$\therefore Y_A = (\alpha)^{X_A} \bmod q$

$= (7)^{69} \bmod 71$

$= 61$

b) $X_B = 15$

$Y_B = (\alpha)^{X_B} \bmod q$

$= (7)^{15} \bmod 71$

$= 23$

c) ~~$K = (Y_A)$~~

$K_1 = (Y_B)^{X_A} \bmod q$

$= (23)^{69} \bmod 71$

$= 34$

$K_2 = (Y_A)^{X_B} \bmod q$

$= (61)^{15} \bmod 71$

$= 34$

$\therefore K = K_1 = K_2 = 34$

5. (RSA)

(a)

i) $n = 84773093$

$\phi(n) = 84754668$

We know in RSA, $n = p * q$

(p prime, q prime)

$\phi(n) = (p-1)(q-1)$

$= pq - p - q + 1$

Now, $n = pq$

$\phi(n) = n - p - \frac{n}{p} + 1$

$\Rightarrow p\phi(n) = pn - p^2 - n + p$

$\Rightarrow p^2 - (n - \phi(n) + 1)p + n = 0$ ——— (1)

There are two solutions of p in eq. (1) and (p, q) are the solutions. This way if n and $\phi(n)$ is given you can compute p and q

Given, $n = 8477\,3093$, $\phi(n) = 84754668$

From eq (1)

$p^2 - (84773093 - 84754668 + 1) \cdot p + 84773093 = 0$

~~$\Rightarrow p = 9539, 8$~~

$\Rightarrow p^2 - 18426p + 84773093 = 0$

$\Rightarrow p = 9539, 8887$

$\therefore (p, q) = (9539, 8887)$

ii) Is there any need to compute p and q by adversary ?

Ans: To decrypt message you need d.

and $d = e^{-1} \bmod \phi(n)$ —— (3)

So, if you know $\phi(n)$, d can be computed using eq. (3). but it is computationally heavy

∴ So, no need to compute p and q

But knowing p and q can help in computing d faster by making use of chinese remainder theorem

∴ adversary can find p and q from $\phi(n)$ to compute d faster

9.

(a) $q = 71, \alpha = 7$

B has public key $Y_B = 3$

Sender A choosed random integer $X_A = 2$.

What is ciphertext of $M = 30$

$K = (Y_B)^K \bmod q = (3)^2 \bmod 71 = 9$

$C_1 = (\alpha)^K \bmod q = (7)^2 \bmod 71 = 49$

$C_2 = KM \bmod q = (9 \times 30) \bmod 71 = 57$

∴ Ciphertext = 57

Fermat's Primality Test

Is p prime?

p is prime if $(a^p - a)$ is a multiple for 'p' for all $1 \le a < p$

Q1) Is 5 prime?

$1 \le a < 5$

$a = 1 \quad 1^5 - 1 = 1 - 1 = 0$

$a = 2 \quad 2^5 - 2 = 32 - 2 = 30$

$a = 3 \quad 3^5 - 3 = 243 - 3 = 240$

$a = 4 \quad 4^5 - 4 = 1024 - 4 = 1020$

} is divisible by $p = 5$

$\therefore p = 5$ is prime

Miller-Rabin Primality Test

n is given prime

Algo

Step 1 $n - 1 = 2^k \times m$

Step 2 Choose 'a' such that $1 < a < n-1$

Step 3 compute $b_0 = a^m \pmod n$, ..., $b_i = b_{i-1}^2 \pmod n$

$+1 \longrightarrow$ composite

$-1 \longrightarrow$ probably prime

9) Is 561 prime

Soln

Given $n = 561$

Step 1

$n-1 = 2^k * m$

$560 = 2^4 \times 35$

$\frac{560}{2^1} = 280 \mid \frac{560}{2^2} = 140$

$\frac{560}{2^3} = 70 \mid \frac{560}{2^4} = 35$

So, $k=4$, $m=35$

Step

Compute $b_0 = a^m \pmod n$

$1 < a < 561-1$

$\Rightarrow 1 < a < 560$

Lets take $a = 2$

$b_0 = 2^{35} \pmod{561}$

$= 263$

$b_0 \neq \pm 1 \pmod{561}$

So, calculate b_1

$b_1 = b_0^2 \pmod{561}$

$= 263^2 \pmod{561}$

$= 166 \neq \pm 1 \pmod{561}$

So, calculate b_2

$b_2 = b_1^2 \pmod{561}$

$= 166^2 \pmod{561}$

$= 67 \neq \pm 1 \pmod{561}$

So, ——— b_3

$b_3 = b_2^2 \pmod{561}$

$= 67^2 \pmod{561} = 1 \longrightarrow$ composite

Solovay - Strassen Algorithm

SOLOVAY_STRASSEN(n)

Choose a random integer a such that $1 \leq a \leq n-1$

$x \leftarrow (a/n)$

If $x = 0$

then return ("n is composite")

$y \leftarrow a^{(n-1)/2} \pmod n$

If $x \equiv y \pmod n$

then return ("n is prime")

else return ("n is composite")

(*) The decision problem is "Is n composite?"

Note that whenever the algorithm says "yes", the answer is correct.

Error may occur when the answer is "no" and the error probability is at most $1/2$.

AKS Primality Test

Let $n \geq 2$, $n \in \mathbb{N}$. Then n is prime if and only if n divides each of the coefficients of

$(x+1)^n - (x^n+1)$

$$(x+1)^n - (x^n+1)$$

$$= \sum_{r=0}^{n} \binom{n}{r} x^r - (x^n+1)$$

$$= \sum_{r=1}^{n-1} \binom{n}{r} x^r$$

n is prime $\iff n \mid \binom{n}{r} x^r$, $r \to 0$ to $n-1$

Eg $(n=3)$

$$(x+1)^3 - (x^3+1)$$

$$= x^3 + 3x^2 + 3.1.x + 1 - x^3 - 1$$

$$= 3x^2 + 3x$$

both div by 3

Hence 3 is prime

Lamport Shostak Peace Algorithm (Agreement Protocol)

n → total

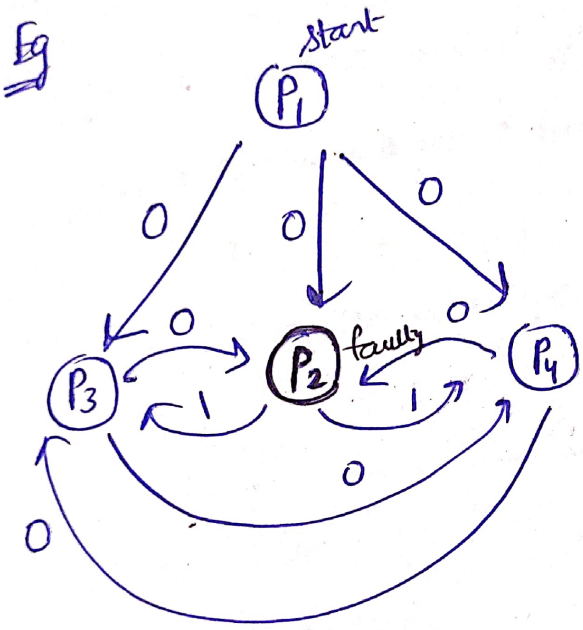
m → faulty

N/W → non-faulty if $m \leq \left\lfloor \frac{n-1}{3} \right\rfloor$

If n=4 $m \leq \left\lfloor \frac{4-1}{3} \right\rfloor$

$m \leq 1$.

Eg



majority

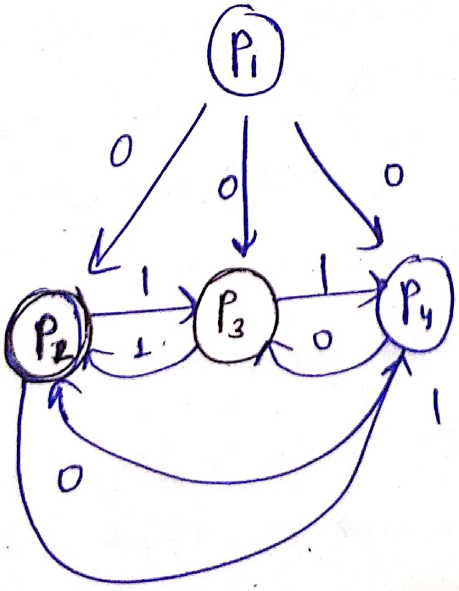
$P_3 = \{0, 1, 0\} = 0$

$P_2 = \{0, 0, 0\} = 0$

$P_4 = \{0, 0, 1\} = 0$

So, System non-faulty

Eg



$P_2 = \{0, 1, 0\} = 0$

$P_3 = \{0, 1, 0\} = 0$

$P_4 = \{1, 1, 0\} = 1$.

So, System faulty

Ceaser / Shift Cipher

$C_i = (P_i + K) \bmod n$

$P_i = (C_i - K) \bmod n.$

Break is easy, just need a bruteforce attack.

Eg: In English Language, 26 letters. So, key 'K' can be any of these 26 letters.

∴ Need to check with 26 keys which one is correct.

Chosen Plaintext attack

↳ Only a character encryption is enough.

COZ we would get C for corresponding P.

$\therefore K = (C - P).$

∴ we would get key (K) i.e shift.

Monoalphabetic Substitution

$C_i = \Pi(m_i)$

$m_i = \Pi^{-1}(C_i)$

Break is easy, just need a frequency analysis. Real world character freq ⟶ cipher text character freq mapping

Chosen Plaintext Attack

To know the character mapping $\begin{cases} A \dots Z \\ 0 \dots 25 \end{cases}$ we have to know the corresponding substitute alphabet for each character

To achieve that, we have to make a plaintext with 26 letters each unique (A..Z). Such that we get mapping for each alphabet.

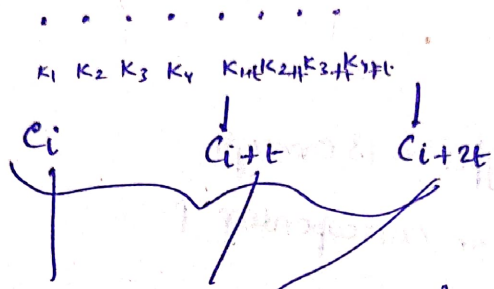
Vigenere Cipher

P.T. a b c d e f g h i j k l m

key w x y z w x y z w x y z w

Now Similar like shift cipher.

Break → we have to know key size. So, we have two info in hand. key size and cipher-text.



All these encrypted by the same key.

So, we have to check 26 keys to check which one is correct.

That would take 26^t tries.

But, we can do statistical analysis.

build frequency table of characters and then check which of the 26 possible shifts gives right prob. distribution 't' frequency tables

Chosen PlainText Attack: If key length is known, then create a plaintext of 't' letters. (need not be unique) we would be able to crack key. If key length not know create plaintext of $O(t_{max})$ → max possible length of key

a a a a a

+ K1 K2 K3 K4 K5

Q. Every encryption scheme for which the size of the key space is equal to the size of the message space, and for which key are chosen uniformly from the key space, is perfectly secret.

(4a) False

Let $M = \{a, b\}$

$K = \{K_1, K_2\}$

$b = \{0, 1\}$

Let $\left.\begin{array}{l} Enc(K, a) = 0 \\ Enc(K, b) = 1 \end{array}\right\}$ for K_1, K_2

Dec. algorithm will return a on input ciphertext 0 and b on input ciphertext 1.

$$Pr[C = 1 | M = a] = 1 \neq 0 = Pr[C = 1 | M = b].$$

Showing that the scheme is not perfectly secret.

(4b) Consider a scheme for shift cipher where only a single character is encrypted. Perfectly secret or not.

$$P(M = m | C = c) = \frac{P[M = m \cap C = c]}{P[C = c]}$$

Encrypt a single letter

26^2 comb. for (p, c). $c = F(p)$

Each p can be mapped to 26 c

26 K's $\quad 26 \times 26 = 26^2$

Now, a message mapped into a ciphertext $\quad P = \frac{1}{26^2}$

$$= \frac{\frac{1}{26^2} \left[\begin{array}{c}\text{out of } 26^2 \\ 1\end{array}\right]}{\frac{1}{26} \left[\begin{array}{c}\text{out of 26 possibilities} \\ 1\end{array}\right]}$$

$$= \frac{1}{26} = P[M = m]$$

$\therefore P[M = m | c = c] = P[M = m]$ (Perfectly Secure).

(4c) Prove or Refute: For every encryption scheme that is perfectly secret, it holds that for every distribution over the message space $\mathcal{M}$, every $m, m' \in M$ and every $c \in C$.

$$\Pr[M=m \mid C=c] = \Pr[M=m' \mid C=c]$$ ✗

Let $M = \{a, b\}$

$\Pr[M=a] = 1/4$

$\Pr[M=b] = 3/4$

$$\Pr[M=a \mid C=c] = \Pr[M=a] = \frac{1}{4}$$

$$\Pr[M=b \mid C=c] = \Pr[M=b] = 3/4$$) ≠

✗

(4d) One Time pad is perfectly ~~Set~~ Secure

(Proof)

⇒ Proof using indistinguishability

$$\Pr[Y=y \mid X=x] = \Pr[X=x, K=k \mid X=x]$$

$$= \Pr[K=k] = \frac{1}{2^L}$$ (Key is of length L So

$Y = X \oplus K$

$$\Pr[Y=y \mid X=x_1] = \frac{1}{2^L}$$

$$\Pr[Y=y \mid X=x_2] = \frac{1}{2^L}$$) =

$\forall x_1, x_2 \in X$

Limitations of Perfect Secrecy

→ key must be as long as the message

→ key must be changed for every encryption

$x_1 \oplus K = y_1$

$x_2 \oplus K = y_2$

$y_1 \oplus y_2 = x_1 \oplus x_2$

Attacker can perform lang analysis to determine y_1, y_2

(Q5) key = length 1 or 2 with 50% prob.

$Pr[M = 'aa'] = 0.4$

$Pr[M = 'ab'] = 0.6$

$Pr[M = 'aa' \mid C = 'bb']$

Given C = bb

Key length 1

M = 'aa' C = 'bb' K = b P = 1/26

M = 'ab' C = 'bb' ~~K = 0~~ no key P = 0

Key length 2

M = 'aa' C = 'bb' K = bb P = $\frac{1}{26} \cdot \frac{1}{26}$

M = 'ab' C = 'bb' K = ba P = $\frac{1}{26} \cdot \frac{1}{26}$

$Pr[C = 'bb']$

$= Pr[M = 'aa'] Pr[C = 'bb' / M = 'aa']$

$+ Pr[M = 'ab'] Pr[C = 'bb' / M = 'ab']$

$= 0.4 \times \frac{1}{2}\left[\frac{1}{26} + \frac{1}{26} \cdot \frac{1}{26}\right]$

$+ 0.6 \times \frac{1}{2}\left[0 + \frac{1}{26} \cdot \frac{1}{26}\right]$

$= 0.0084$

$\therefore Pr[M = 'aa' \mid C = 'bb'] = \frac{Pr[M = 'aa' \cap C = 'bb']}{Pr[C = 'bb']}$

$= \frac{Pr[M = 'aa'] Pr[C = 'bb' / M = 'aa']}{Pr[C = 'bb']}$

$= \frac{0.4 \times \frac{1}{2}\left[\frac{1}{26} + \frac{1}{26} \cdot \frac{1}{26}\right]}{0.0084}$

$=$

(6)

Perfect Distinguishability Experiment $PrivK^{eav}_{A,\Pi}$

→ Adversary A chooses two messages $m_0, m_1 \in M$ not necessarily of same length

→ Bob generates a key $k \leftarrow Gen$ and a bit $b \leftarrow \{0,1\}$ He computes and gives the ciphertext $c \leftarrow Enc_k(m_b)$ to A.

→ A outputs a bit b', trying to tell whether c is the encryption of m_0 or m_1

→ The output $PrivK^{eav}_{A,\Pi}$ of the experiment is 1 iff $b = b'$ (i.e. A succeeds)

Definition of perfect indistinguishability

→ Encryption scheme $\Pi = (Gen, Enc, Dec)$ with message space M

→ Adversary: An eavesdropper with unlimited computing power

→ We model the adversary as a probabilistic algorithm A that on input $m_0, m_1 \in M$ and $c \in C$ outputs a bit $b \in \{0,1\}$

→ An encryption scheme is perfectly indistinguishable

If for every A and every two $m_0, m_1 \in M$

$$Pr\left[PrivK^{eav}_{A,\Pi}(m_0, m_1) = 1\right] = \frac{1}{2}$$

Perfect Secrecy ⇒ Perfect indistinguishability

- If the encryption scheme is perfectly secret then

$$Pr[Enc_K(m_0) = c] = Pr[Enc_K(m_1) = c] \quad \forall\ m_0, m_1 \in M, c \in C$$

$$\rightarrow Pr[PrivK^{eav}_{A,\Pi}(m_0, m_1) = 1] \quad (= Pr[A \text{ wins}])$$

$$= \frac{1}{2} \sum_{i=0,1; c \in C} Pr[b = i \wedge Enc_K(m_i) = c \wedge A(m_0, m_1, c) = i]$$

$$= \sum_{c \in C} \sum_{i=0,1} Pr[b = i] \cdot Pr[Enc_K(m_i) = c] \cdot Pr[A \neq (m_0, m_1, c) \neq i]$$

$$= \frac{1}{2} \sum_{c \in C} \left(Pr[Enc(m_0) = c] \cdot \sum Pr[A(m_0, m_1, c) = i] \right)$$

$$= \frac{1}{2}$$

(7a) **Negligible Function** :- A function f is negligible if for every polynomial $p(\cdot)$ there exists an N such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$

eg $\frac{1}{2^n}$; $\frac{1}{n!}$

(7b) i) $\frac{1}{2^n}$ (negligible)

ii) $\frac{1}{(\log n)!}$ (negligible)

iii) $\frac{1}{(\log\log n)!}$ (negligible)

iv) $\frac{1}{10^{10}}$ not negligible bcoz constant fn

v) $n^{1/n}$ (not negligible bcoz exp func)

vi) $\frac{1}{2}$ (not negligible bcoz const fn)

vii) $\frac{n}{2^n}$ (negligible bcoz in denominator exp fn.)

viii) $\frac{1}{n}$ (Should not be negligible bcoz in denominator polynomial fn only)

Stirlings appx

$(\log n)! \cong \leq e(\log n)^{(\frac{1}{2} + \log n)}$

so $(\log n)!$ grows faster than polynomial * $e^{(-\log n)}$

7c) We have to prove $h(n) \leq n^{-c}$

$f(n), g(n) \leq n^{-(c+1)}$

We have to prove $h(n) \leq f(n)+g(n) \leq n^{-c}$

Then since $c+1 \in \mathbb{N}$ and since f and g are negligible there exists n_f and n_g such that:

$\forall n > n_f, \; f(n) \leq n^{-(c+1)}$

$\forall n > n_g, \; g(n) \leq n^{-(c+1)}$

Choose $n_0 = \max(n_f, n_g, 2)$

Then for any $n \geq n_0$. we have

$$h(n) = f(n)+g(n)$$
$$\leq n^{-(c+1)} + n^{-(c+1)}$$
$$\leq 2n^{-(c+1)}$$
$$\leq n \cdot n^{-(c+1)} \quad (\text{since } n \geq n_0 \geq 2)$$

Since, $n^{-c} < n \cdot n^{-(c+1)}$

$\therefore h(n) \leq n^{-c}$

Hence $h(n)$ is also negligible.

8

a) PRG

Defn: Let $l(\cdot)$ be a polynomial and G be a deterministic polynomial-time algorithm such that for any input $s \in \{0,1\}^n$ algorithm G outputs a string of length $l(n)$. We say G is a PRG if following conditions hold:-

1. for every n it holds that $l(n) > n$ (Expansion)

2. (Pseudorandomness): For all probabilistic polynomial time distinguisher D, there exists a negligible func negl s.t.

$$\left| Pr[D(r)=1] - Pr[D(G(s)=1] \right| \leq negl(n)$$

where r is chosen uniformly at random from $\{0,1\}^{l(n)}$

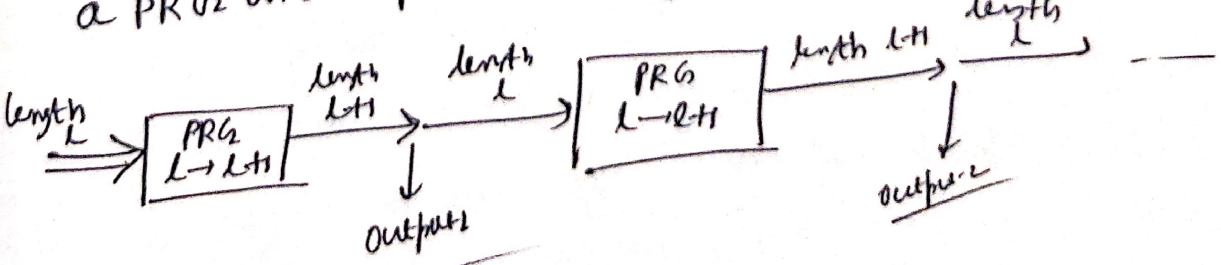
(PRG) ← seed s is chosen from $\{0,1\}^n$

↓

(truly random generator)(TRG)

Expansion in PRG

Assume that there exists a PRG with expansion factor $l(n) = n+1$. Then for any polynomial $p(\cdot)$, there exists a PRG with expansion-factor $l(n) = p(n)$



(Q9) One-way func :- ——— defn + Example + ∄ explanation

(a) $h(x) \stackrel{def}{=} f(x) \oplus g(x)$

Not one-way

If $f(x) = g(x)$ then $h(x) = 0$

Then all $h(x)$ maps to '0'

∴ One-way ✗

(b) $h(x) \stackrel{def}{=} f(x) \,||\, g(x)$

#

(c)

(Q10)

(Q11)

(Q12) On input an $(n+1)$-bit string y, output 0 if the last bit of y is 0. The last bit of $G_2(s)$ is always 0, in constrast to a random string where the last bit is 0 only half the time

(Q13)

$G_2(x) = F_x(0\ldots0)$ where x is a 128 bit i/p

$G_2(x) = F_x(0\ldots0) \,||\, F_x(1\ldots1)$, where x is a 128 bit i/p

(Q14)

(Q15)

(a) Any private key encryption scheme i.e. CPA Secure must also be computationally indistinguishable (T)

(b) Any private key encryption scheme that is CCA Secure must also be perfect Secret (T)

(c) private-key CCA Secure $\Rightarrow$ CPA Secure (T)

(Q16)

Option d Let m_1 and m_2 be arbitrary but distinct. Using the encryption oracle, obtain an encryption, obtain an encryption $r \| c_1 \| c_2$ of $m_1 \| m_2$. O/p messages $M_0 = m_1 \| m_2$ and $M_1 = m_2 \| m_1$. Output 0 if the third block of the challenge ciphertext is c_2.

(Q17)

One-way Func

A function $f: \{0,1\}^* \longrightarrow \{0,1\}^*$ is one-way if following two conditions hold.

1. Easy to compute: There exist a polynomial time algorithm M_f computing f; that $M_f(x) = f(x) \; \forall x$

2. Hard to invert: For every probabilistic polynomial time algo there exist a negligible fn such that

$$Pr[\text{Invert}_{A,f}(n) = 1] \leq \text{negl}(n)$$

$$f_{n,g}(x) = g^x \bmod n$$

One-way permutation

A function f is a one-way permutation if for every n, f restricted to $\{0,1\}^n$ is a permutation, and for all polynomials $s(n)$ and $\forall$ sufficiently large n, f is $s(n)$ secure