Mod - 2

Cryptographic Arithmetic.

↳ It provides the mathematical foundation for encryption algorithms. It includes modular arithmetic, inverses, and Euclidean algorithm.

1] Modular Arithmetic.

↳ It involves number wrapping around after reaching a certain modulus m.

e.g $12 \bmod 7 = 5$.

2] Additive and Multiplicative Inverse.

↳ a number b is an additive inverse of a if $a+b \equiv 0 \bmod m$

e.g $a=3 \quad m=7 \quad b=4$ because $3+4 \equiv 0 \bmod 7$

Multiplicative Inverse.

a number b is a Multiplicative inverse of a if $a * b \equiv 1 \bmod m$

e.g $3 \times b \equiv 1 \bmod 7$ trying $b=5$

$3 \times 5 = 15 \bmod 7 \equiv 1$

Euclidean Algorithm.

Note when gcd(a,b) = 1, a and b are termed as Relatively prime.

Example. For Extended euclidean Algorithm.

a = 161 b = 28 find GCD (a,b).

Q = R1/R2	R1	R2	R	S1	S2	S	T1	T2	T
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

$S = S_1 - Q \times S_2$ $T = T_1 - Q - T_2$.

Initially $S_1 = 1$ $S_2 = 0$ $T_1 = 0$ $T_2 = 1$.

GCD = 7 = R_1

Note If the GCD of two number is 1 then only we can find inverse key for b which is t_1.

Mathematics for Asymmetric Key Cryptogra

1] Prime Generation:

↳ In asymmetric key cryptographic systems (e.g. RSA, Diffie-Hellman, ECC), large prime Number are essential.

Steps for generating Cryptographic Primes.

→ Random Number Selection

A large Random odd Number is chosen, typically 1024-bit, 2048-bit or higher.

→ Trial Division

The number is checked for divisibil against small prime Numbers (e.g 2, 3, 5, 7 ... to eliminate obvious composites.

→ Probabilistic Primality Testing

If the Number passes trivial division, it undergoes probabilistic tests like Miller-Rabin or Fermat's Test.

→ Strong Prime Selection

In some cryptographic applications (e.g RSA), strong prime (Primes whose neighbors are also hard to factor) are preferred.

2] Primality Testing.

↳ Since directly checking if a number is prime is computationally expensive.

→ Miller - Rabin Test.

- Given a number n, it is tested for primality using random bases a.
- If n fails for any bases, It is composite
- If n passes multiple rounds, it is probably prime.

→ Fermat's Primality Test.

- Based on Fermat's ~~#~~ Little Theorem. $(a^{p-1} \equiv 1) \bmod p$

Deterministic Primality Tests.
(Slower but always correct).

1. AKS Primality Test
   - A polynomial - time deterministic algorithm.
   - Guarantees primality but is too slow for practical cryptography.
2. Elliptic Curve primality Proving (ECPP).
   - Used to prove that a Numbe is prime.

3] Prime Factorization.

↳ It is a process of breaking dow a Number into its prime component.

e.g $n = 91 \Rightarrow 91 = 7 \times 13$.

- Why is Prime Factorization is Important?

1. RSA Cryptography.
   - RSA relies on the difficulty of factoring large Number.
   - Given $n = p \times q$ finding p and q is Computationally Infeasible.

4] Euler's Totient Function

↳ It counts the number of integers less than n that are coprime to n.

$\phi(p) = p-1$ if $n = P \times q$ then $\phi(n) = (P-1) \times (q$

Role in Cryptography.

- In RSA encryption, the Totient function is used to Compute private key.

$d \equiv e^{-1} \pmod{(\phi(n))}$.

Key Management.

↳ It refers to the process involved in handling Cryptographic keys, including their generation, storage, distribution, and disposal.

The lifecycle of cryptographic key.

1. Generation - Creating strong and cryptographic key.
2. Distribution - Securely delivering key to authorized entities.
3. Storage - Safeguarding key against unauthorized access.
4. Usage - Using key for encryption, decryption and signing etc.
5. Rotation - Changing key periodically to prevent compromise.
6. Revocation - Disabling compromised key
7. Destruction - Securely deleting expired or unused key.

Diffie - Hellman.

Algorithm

Step 1 Choosing a Prime Number
Step 2 Finding primitive Root.
Step 3 Selecting private key.
Step 4 Generating public key.
Step 5 Computing the shared secret.

example

$p = 23$

Choose primitive Root g let $g = 5$.

private key $X_a = 6$
private key $X_b = 15$

Compute Public key.

$Y_a = 5^6 \bmod 23.$
$\therefore Y_a = 8$
Ily $Y_b = 5^{15} \bmod 23$
$= 19.$

Compute the shared Secret key

$K_a = Y_a^{X_a} \bmod 23 = 19^6 \bmod 23 \quad K_a = 2$
$K_b = Y_a^{X_b} \bmod 23 = 8^{15} \bmod 23 \quad K_b = 2.$

Strength

- Secure key exchange Over Insecure Channel.
- Forward Secreacy.
- Based on Hard Mathematical problem.

Weaknesses

- Velnerability to Man-in-the-middle attack
- No Built in authentication
- Parameter Sensitive.

Applications.

- VPN - Secure Connection in IPsec
- Messaging Apps - Powers end-to-end encryption in whatsapp.
- Blockchain & Cryptocurrencies - Secure Bitcoin wallets and Transaction.