## ElGamal Encryption *(in Brief)*

Understanding Public-Key Cryptography with an Example

- Introduction to ElGamal Encryption
- Importance in cryptography
- Overview, Key Generation, Encryption, Decryption, Full Example

### What is ElGamal Encryption?

- A public-key cryptosystem based on the Diffie-Hellman Key Exchange.
- Developed by: Taher ElGamal in 1985.
- Uses discrete logarithms for security.
- Applications: Secure communication, digital signatures, etc.

### Key Features of ElGamal Encryption

1. Asymmetric: Uses a pair of public and private keys.
2. Based on Modular Arithmetic: Relies on mathematical properties of large prime numbers.
3. Probabilistic: Produces different ciphertexts for the same plaintext.
4. Security: Depends on the difficulty of solving the discrete logarithm problem.

### Key Generation Process

1. Choose a large prime number p and a generator g.
2. Select a private key x, where $x \in [1, 2, \ldots, p-2]$.
3. Compute the public key y using $y = g^x \bmod p$.
4. Share (p, g, y) as the public key. Keep x secret as the private key.

## Example:

- $p = 23$, $g = 5$
- Private key: $x = 6$
- Public key: $y = 5^6 \bmod 23 = 8$

Public Key: ( $p = 23$, $g = 5$, $y = 8$ )

Private Key: $x = 6$

## Encryption Process

1. Sender chooses a random integer k such that $k \in [1, 2, \ldots, p-2]$.
2. Compute $c_1 = g^k \bmod p$.
3. Compute $c_2 = m \cdot y^k \bmod p$, where m is the plaintext.
4. Send ciphertext: $(c_1, c_2)$.

## Example:

- Plaintext: $m = 7$, $k = 3$
- $c_1 = 5^3 \bmod 23 = 10$
- $c_2 = 7 \cdot 8^3 \bmod 23 = 21$

  Ciphertext: $(c_1 = 10, c_2 = 21)$

## Decryption Process

1. Compute $s = c_1^x \bmod p$, where x is the private key.
2. Compute the inverse of s, denoted $s^{-1}$, modulo p.
3. Recover plaintext: $m = c_2 \cdot s^{-1} \bmod p$.

## Example:

- $s = 10^6 \bmod 23 = 9$
- $s^{-1} = 9^{-1} \bmod 23 = 18$
- $m = 21 \cdot 18 \bmod 23 = 7$

  Recovered plaintext: $m = 7$.

## Full Example Recap

- Setup: $p = 23$, $g = 5$, $x = 6$, $y = 8$.
- Encryption:
  - Plaintext: $m = 7$, $k = 3$.
  - Ciphertext: $(c_1=10, c_2=21)$.
- Decryption:
  - $s = c_1^x \bmod p = 9$.
  - $s^{-1} = 18$.
  - Recovered $m = 7$.

## Advantages and Limitations

### Advantages:

1. Strong security based on discrete logarithms.
2. Randomization makes it secure against chosen-plaintext attacks.

### Limitations:

1. Computationally intensive.
2. Ciphertext size is larger than plaintext size.

## Applications

- Secure Communication: Ensures confidentiality in messages.
- Digital Signatures: Forms the basis for many signature schemes.
- Cryptographic Protocols: Used in hybrid encryption systems.

## Conclusion

- Summary: ElGamal is a secure, robust encryption scheme suitable for various cryptographic applications.
- Takeaway: Importance of understanding modular arithmetic and key management.
- Next Steps: Explore implementation using programming languages (e.g., Python).