ElGamal Encryption *(in Details)*

1. Introduction to ElGamal Encryption

ElGamal encryption is a public-key cryptosystem introduced by Taher ElGamal in 1985. It is widely used in cryptographic protocols and provides security based on the computational difficulty of the discrete logarithm problem (DLP). ElGamal encryption is commonly employed in secure communication systems, including digital signatures and encryption schemes.

2. Overview of Public-Key Cryptography

Public-key cryptography involves:

1. **Key Generation**: The generation of a pair of keys:
    - A **public key** for encryption, shared with others.
    - A **private key** for decryption, kept secret.
2. **Encryption**: Using the recipient's public key to encode a message.
3. **Decryption**: Using the recipient's private key to decode the message.

ElGamal encryption is asymmetric and provides the following key benefits:

- Confidentiality: Ensures only authorized parties can read the message.
- Non-repudiation: Prevents the sender from denying the transmission.

3. Key Components of ElGamal Encryption

ElGamal encryption uses the following:

1. **Prime Number p**: A large prime number.
2. **Generator g**: A primitive root modulo p.
3. **Private Key x**: A random number such that $1 \leq x < p$.
4. **Public Key y**: Computed as $y = g^x \bmod p$.

4. The ElGamal Encryption Process

Step 1: Key Generation

1. Choose a large prime number p.
2. Select a generator g modulo p.
3. Choose a private key x, a random integer where $1 \leq x < p-1$.
4. Compute the public key $y = g^x \bmod p$.

The public key is (p, g, y), and the private key is x.

Step 2: Encryption

To encrypt a message m:

1. Represent the message m as an integer $0 \leq m < p$.
2. Select a random ephemeral key k, where $1 \leq k < p-1$.
3. Compute:
   - $c_1 = g^k \bmod p$,
   - $c_2 = (m \cdot y^k) \bmod p$.
4. The ciphertext is (c_1, c_2).

Step 3: Decryption

To decrypt the ciphertext (c_1, c_2):

1. Compute the shared secret $s = c_1^x \bmod p$.
2. Recover the plaintext m:
   - $m = (c_2 \cdot s^{-1}) \bmod p$, where s^{-1} is the modular inverse of s modulo p.

5. Full Example of ElGamal Encryption and Decryption

Key Generation

1. Select p=23 (a prime number).
2. Choose g=5 (a primitive root modulo 23).
3. Private key x=6.
4. Compute the public key: $y = g^x \mod p = 5^6 \mod 23 = 15$.
   Public key: (p, g, y) = (23, 5, 15).

Encryption

Let the message m=13.

1. Choose a random ephemeral key k=3.
2. Compute:
   - $c_1 = g^k \mod p = 5^3 \mod 23 = 10$,
   - $c_2 = (m \cdot y^k) \mod p = (13 \cdot 15^3) \mod 23$.

   First, compute $15^3 \mod 23$:

   $15^3 = 3375, \quad 3375 \mod 23 = 21$.

   Then:

   $c_2 = (13 \cdot 21) \mod 23 = 273 \mod 23 = 20$.

   Ciphertext: $(c_1, c_2) = (10, 20)$.

Decryption

To decrypt $(c_1, c_2) = (10, 20)$:

1. Compute the shared secret $s = c_1^x \bmod p$:

   $s = 10^6 \bmod 23$.

   Compute step-by-step:

   $10^2 = 100 \bmod 23 = 8$, $10^4 = 8^2 \bmod 23 = 64 \bmod 23 = 18$,

   $10^6 = 18 \cdot 8 \bmod 23 = 144 \bmod 23 = 6$.

   Thus, s=6.

2. Compute $s^{-1} \bmod p$, the modular inverse of $s = 6$ modulo 23:
   Using the extended Euclidean algorithm, $s^{-1} = 4$.
3. Recover m:

   $m = (c_2 \cdot s^{-1}) \bmod p = (20 \cdot 4) \bmod 23 = 80 \bmod 23 = 13$.
   Decrypted message: m=13.

6. Advantages of ElGamal Encryption

1. **Security**: Based on the discrete logarithm problem, providing strong cryptographic guarantees.
2. **Flexibility**: Supports various cryptographic protocols.
3. **Randomness**: Ephemeral keys k ensure that identical plaintexts encrypt to different ciphertexts.

7. Limitations of ElGamal Encryption

1. **Ciphertext Expansion**: The ciphertext size is double the plaintext size.
2. **Efficiency**: Computationally intensive compared to symmetric cryptography.
3. **Ephemeral Key Management**: Secure generation and storage of k is critical.

8. Applications of ElGamal Encryption

1. Secure communication protocols (e.g., SSH, TLS).
2. Digital signature schemes (e.g., Digital Signature Algorithm (DSA)).
3. Cryptographic voting systems and key exchange protocols.

9. Conclusion

ElGamal encryption is a foundational public-key cryptosystem offering robust security through the discrete logarithm problem. While its computational complexity and ciphertext expansion pose challenges, its versatility and strong cryptographic guarantees make it a widely respected encryption technique.