Encryption Systems Report

Ihsaan Hussain(210260021)

June 19, 2024

Abstract

This is a report for RSA, Paillier and Elgamal cryptosystems for the Week 1: Python & Cryptography Assignment 2 . The report contains key generation, encryption and decryption techniques corresponding to the three cryptosystems.

1. RSA

RSA involves a pair of keys: a public key for encryption and a private key for decryption. RSA is a public key cryptosystem which is based on the assumption that there is a wide class of functions that are relatively easy to compute but extraordinarily difficult to invert unless you possess a secret.

1.1. Key generation

(I) Generate 2 large prime numbers: p and q

(II) Compute N = p×q

(III) Compute Euler's Totient Function:

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

.

(IV) Choose an Integer e: e (public exponent) such that

$$1 < e < \phi(N)$$

and

$$gcd(e, \phi(N)) = 1$$

The pair (e,N) forms the public key.

(V) Compute d: d (private exponent) such that

$$d \equiv e^{-1} \mod \phi(N)$$

The pair (d,N) forms the private key.

1.2. Encryption

Plaintext m is encrypted by the sender using the public key (e,N).

$$s = m^e \mod N$$

where s is the ciphertext.

1.3. Decryption

The original message is decrypted by the receiver using the private key (d,N).

$$m = s^d \mod N$$

where m is the original message.

1.4. Proof

$$s^d = m^{ed} \mod N$$

Since

$$ed \equiv 1 \mod \phi(N)$$
$$ed = k\phi(N) + 1$$

for some integer k

$$m^{ed} = (m^{\phi(N)})^k m \mod N$$

Now GCD(m,p) = GCD(m,q) = 1 since p,q are primes and m is less than p and q. Applying Fermat's Little Theorem:

$$m^{p-1} \equiv 1 \mod p$$
$$m^{q-1} \equiv 1 \mod q$$

Applying Chinese Remainder Theorem:

$$m^{(p-1)(q-1)} \equiv 1 \mod pq$$

Thus

$$s^d = m^{ed} = 1^k m \mod N$$

which proves that

$$m = s^d \mod N$$

1.5. Security of RSA

The security of RSA encryption relies on the mathematical difficulty of factorizing large composite numbers. The public key is composed of the modulus N and the public exponent e. The modulus N is a product of two large primes, and the security relies on N being large enough to prevent efficient factorization.

1.6. Implementation

The message is split into characters and each character is encrypted and the encypted characters and joined to form the ciphertext. The cipher text is then decrypted to display the original message. The process is done through the following functions:

1. generate 100 digit prime(): Generates 100 digit primes using sympy module.
2. keys(): Generates public and private key. Multiplicative inverse is calculated through extended euclidean theorem and coprime is calculated using euclidean GCD.
3. encrypt(): Encrypts the characters of the message.
4. decrypty(): Decrypts the characters of the message.
5. modularexponentiation(): This function efficiently computes large powers modulo N using the method of exponentiation by squaring, which has a time complexity of O(logb).

When the program is run, the user is asked to input a text message. The encrypted and decrypted text message is then displayed.

2. Paillier

The Paillier cryptosystem is based on the difficulty of computing n-th residues modulo n^2. It is a partially homomorphic encryption scheme that enables computations on encrypted data with respect to addition and multiplication.

2.1. Key Generation

1. **Choose Two Large Prime Numbers:** p and q.
2. **Compute n:** $n = p \times q$.
3. **Compute λ:** $\lambda = (p-1, q-1)$.
4. **Choose g:** g = n+1
5. **Compute μ:** $\mu = \lambda^{-1} \mod n$.
6. **Public Key:** (n, g).
7. **Private Key:** (n, λ, μ).

2.2. Encryption

Plaintext m is encrypted by the sender using the public key (n, g) as follows:

1. **Select a Random r:** Choose $r \in \mathbb{Z}_n^*$.
2. **Compute Ciphertext c:** $c = g^m \cdot r^n \mod n^2$.

2.3. Decryption

The original message is decrypted by the receiver usinf the private key (n, λ, μ) as follows:

1. **Compute Intermediate Value d:** $d = c^\lambda \mod n^2$.
2. **Compute L(d):** L(d) = (d-1)/n
3. **Compute Plaintext m:** $m = L(d) \cdot \mu \mod n$.

2.4. Security of Paillier Cryptosystem

The security of the Paillier cryptosystem is based on the decisional composite residuosity assumption (DCRA), which states that it is computationally infeasible to decide whether a given number is an n-th residue modulo n^2 without knowledge of the factors of n.

2.5. Homomorphic Properties

One of the distinguishing features of the Paillier cryptosystem is its homomorphic properties, allowing the following operations to be performed on ciphertexts:

1. **Homomorphic Addition of Plaintexts:**

$$E(m_1 + m_2, r_1.r_2) = g^{m_1+m_2}.(r_1.r_2)^n (mod n^2)$$

2. **Homomorphic Multiplication of Plaintext with Constant:**

$$E(m_1 m_2, r_1 m_2) = g^{m_1 m_2}.r_1^{n m_2} (mod n^2)$$

2.6. Implementation

The message is split into characters and each character is encrypted and the encypted characters and joined to form the ciphertext. The cipher text is then decrypted to display the original message. The process is done through the following functions:

1. generate large prime(): Generates 7 digit primes using sympy module. Primes larger than 7 digits leads to overflow.

2. keys(): Generates public and private key. Multiplicative inverse is calculated through extended euclidean theorem .

3. encrypt(): Encrypts the characters of the message. coprime is calculated using euclidean GCD

4. decrypty(): Decrypts the characters of the message.

5. modularexponentiation(): This function efficiently computes large powers modulo N using the method of exponentiation by squaring, which has a time complexity of O(logb).

When the program is run, the user is asked to input a text message. The encrypted and decrypted text message is then displayed.

3. Elgamal

The ElGamal cryptosystem is based on the fact that it is easy to raise things to powers in modular arithmetic but difficult to take (discrete) logarithms.

3.1. Key Generation

1. **Choose a Large Prime p:** Select a large prime number p.

2. **Select a primitive root α:** Select a primitive root α of p .

3. **Private Key:** Select a private key a such that $1 < a < p - 1$.

4. **Public Key:** Compute $\beta \equiv \alpha^a \mod p$. The public key is (p, α, β).

3.2. Encryption

To encrypt a message m for a recipient with the public key (p, α, β):

1. **Convert Message to an Integer:** Convert the plaintext message m to an integer P such that $0 < P < p$.

2. **Select a Random Integer k:** Choose a random integer k such that $1 < k < p - 2$.

3. **Compute Ciphertext Pair:**

$$c_1 = \alpha^k \mod p$$

$$c_2 = P \cdot \beta^k \mod p$$

The ciphertext is the pair (c_1, c_2).

3.3. Decryption

To decrypt a ciphertext (c_1, c_2) with the private key a:

1. **Recover the original message:**

$$P = c_1^{p-1-\alpha} c_2 \mod p$$

3.4. Security of ElGamal Cryptosystem

The security of the ElGamal cryptosystem relies on the intractability of the discrete logarithm problem. The ciphertext is secure as long as the discrete logarithm remains a hard problem.

3.5. Homomorphic Properties

ElGamal encryption supports multiplicative homomorphism, meaning that the product of two ciphertexts decrypts to the product of their corresponding plaintexts:

$$D(E(m_1) \cdot E(m_2) \mod p) = m_1 \cdot m_2 \mod p$$

3.6. Implementation

The message is split into characters and each character is encrypted and the encypted characters and joined to form the ciphertext. The cipher text is then decrypted to display the original message. The process is done through the following functions:

1. generate large prime(): Generates 29 digit primes using sympy module. Primes larger than 29 digits leads to overflow.
2. is primitive root(α, p): Checks whether α is a primitive root to p.
3. keys(): Generates public and private key.
4. encrypt(): Encrypts the characters of the message.
5. decrypty(): Decrypts the characters of the message.
6. modularexponentiation(): This function efficiently computes large powers modulo N using the method of exponentiation by squaring, which has a time complexity of O(logb).

When the program is run, the user is asked to input a text message. The encrypted text (c_1, c_2) and decrypted text message is then displayed.