Mustansiriyah University	Public-Key Cryptography	Class: Third Year
Engineering College	Asymmetric Cipher	Course name: Data Encryption
Computer Engineering Dep.	Paillier Algorithm	Lecturer: Fatimah Al-Ubaidy

Paillier Cryptosystem

The Paillier cryptosystem is invented by Pascal Paillier in 1999 and is based on public key cryptography. It is a partial homomorphic encryption scheme which allows two types of computation:

- Addition of two plaintexts ($D_{priv}(E_{pub}(m1) \times E_{pub}(m2) \bmod n^2) = (m1 + m2) \bmod n$) $\Rightarrow c1 \times c2 = m1 + m2$
- Multiplication of a ciphertext by a plaintext number ($D_{priv}(E_{pub}(m1)^{m2} \bmod n^2) = (m1 \times m2) \bmod n$) $\Rightarrow c1^{m2} = m1 \times m2$

Its algorithm consists of three parts: Key generation, encryption, and decryption scheme.

Key generation:

(1) Pick two large prime numbers p and q, randomly and independently. Confirm that $\gcd(p \times q, (p-1) \times (q-1))$ is 1. If not, start again.

(2) Compute $n = p \times q$.

(3) Compute λ as $\text{lcm}(p-1, q-1)$ where lcm(.) means least common multiple.

(4) Pick a random integer g in the set $Z^*_{n^2}$ (integers between 1 and n^2).

(5) Calculate the modular multiplicative inverse $\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n$. If μ does not exist, start again from step 1.

Where the function $L(x) = (x-1) / n$ (quotient of integer division).

The public key is (n,g). Use this for encryption.

The private key is (λ, μ). Use this for decryption.

Paillier Cryptosystem

Paillier Encryption Scheme:

(1) Pick a random number r in the range $0 < r < n$.

(2) Compute the ciphertext $c = g^m \times r^n \bmod n^2$.

Paillier Decryption Scheme:

Compute the plaintext $m = L(c^\lambda \bmod n^2) \times \mu \bmod n$.

Example:

Key generation

(1) Pick $p = 13$ and $q = 17$. (They satisfy the condition.)

(2) Compute $n = 221$.

(3) Compute $\lambda = 48$.

(4) Pick $g = 4886$.

(5) Compute $\mu = 159$. (It exists.)

Encryption

Set $m = 123$.

(1) Pick $r = 59$.

(2) Compute $c = 13250 \bmod 221^2$.

Decryption

Compute $m_{decrypted} = 123 \bmod 221$. (The same as m1.)