# Implementasi Paillier Cryptosystem

Tony - 13516010
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl Ganesha 10 Bandung 40132, Indonesia
13516010@std.stei.itb.ac.id

***Abstract*—Algoritma *Paillier Cryptosystem* merupakan salah satu dari banyak algoritma kunci publik. Meskipun algoritma ini kurang populer, ternyata implementasi dari algoritma ini cukup mudah.**

***Keywords— Paillier Cryptosystem; kriptografi kunci publik; kriptografi; implementasi***

## I. PENDAHULUAN

Terdapat banyak algoritma kriptografi kunci publik. Dari semua itu, algoritma-algoritma yang populer hanyal beberapa, antara lain algoritma *RSA* (*Rivest-Shamir-Adleman*), *Diffie-Hellman*, *Elgamal*, dan *ECC* (*Elliptic Curve Cryptography*). Dari semua algoritma kriptografi kunci publik, dibahas mengenai algoritma *Paillier Cryptosystem*. Selain itu, terdapat implementasi dari algoritma tersebut.

## II. DASAR TEORI

### A. Kriptografi

Kriptografi merupakan ilmu dan seni untuk menjaga keamanan pesan. Kriptografi modern berfokus pada aspek *confidentiality*, *data integrity*, *authentication*, dan *non-repudiation*. Algoritma kriptografi modern dirancang berdasarkan teori-teori matematika dan ilmu komputer. Kriptografi modern terbagi atas kriptografi kunci simetris dan kriptografi kunci publik.

### B. Kriptografi Kunci Publik

Kriptografi kunci publik atau kriptografi asimetris adalah sistem kriptografi yang menggunakan sepasang kunci, yaitu kunci publik dan kunci privat. Kunci publik merupakan kunci yang dapat disebar luaskan. Berbeda dari kunci publik, kunci privat adalah kunci rahasia yang hanya diketahui oleh pemiliknya. Operasi enkripsi menggunakan salah satu dari kunci tersebut dan operasi dekripsi menggunakan kunci lainnya. Salah satu keuntungan dari kriptografi kunci publik adalah tidak perlu adanya mekanisme pengiriman kunci rahasia.

### C. Paillier Cryptosystem

*Paillier cryptosystem* diciptakan oleh Pascal Paillier pada tahun 1999. *Paillier cryptosystem* merupakan kriptografi kunci publik yang probabilistik. Pada algoritma paillier, kunci publik digunakan untuk mengenkripsi data dan kunci privat digunakan untuk mendekripsi data. Algoritma pembentukan kunci dari *Paillier* adalah sebagai berikut.

1. Tentukan 2 buah bilangan prima $p$ dan $q$ secara acak, sehingga memenuhi $fpb(pq, (p-1)(q-1)) = 1$. Disini, $fpb$ berarti faktor persekutuan terbesar. Persamaan tersebut dijamin terpenuhi jika panjang $p$ dan $q$ sama.
2. Hitung $n = pq$ dan $\lambda = kpk(p-1, q-1)$. Disini, $kpk$ adalah kelipatan persekutan terkecil.
3. Pilih sembarang bilangan bulat $g$, yang mana $g \in \mathbb{Z}^*_{n^2}$.
4. Pastikan bahwa $n$ habis membagi order dari $g$, dengan melakukan pengecekan terhadap rumus $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$, dimana fungsi $L$ adalah $L(x) = \frac{x-1}{n}$.
5. Kunci publik berupa $(n, g)$ dan kunci privatnya adalah $(\lambda, \mu)$.

Algortima untuk melakukan enkripsi adalah sebagai berikut.

1. Misalkan m merupakan pesan yang akan dienkripsi, yang memenuhi $0 \leq m < n$.
2. Tentukan bilangan bulat $r$ secara acak, yang mana $0 < r < n$ dan $r \in \mathbb{Z}^*_n$.
3. *Ciphertext* $c$ dapat dihitung dengan rumus $c = g^m * r^n \bmod n^2$.

Algoritma untuk melakukan dekripsi adalah sebagai berikut.

1. Misalkan $c$ merupakan *ciphertext*, yang mana $c \in \mathbb{Z}^*_{n^2}$.
2. *Plaintext* $m$ dapat dihitung dengan rumus $m = (L(c^\lambda \bmod n^2) * \mu) \bmod n$.

*Paillier cryptosystem* memiliki sifat *partially homomorphic encryption* yang mendukung operasi adiktif. Dari sifat tersebut dapat didefinisikan identitas berikut.

1. Penjumlahan dari *plaintext*

Perkalian dari dua buah *ciphertext* akan menghasilkan penjumlahan dari *plaintext*-nya.

$$D(E(m1, r1) * E(m2, r2) \bmod n^2) = m1 + m2 \bmod n$$

2. Perkalian dari *plaintext*

   *Dekripsi dari ciphertext* yang dipangkatkan dengan *plaintext* akan menghasilkan perkalian kedua *plaintext*.

$$D(E(m1, r1)^{m2} \bmod n^2) = m1 * m2 \bmod n$$

### D. Enkripsi Homomorfik (Homomorphic Encryption)

Enkripsi homomorfik adalah algoritma enkripsi yang memungkinan untuk dilakukannya komputasi terhadap data terenkripsi atau *ciphertext*. Secara matematis, enkripsi homomorfik melakukan transformasi dari suatu himpunan data menjadi himpunan data yang lain dengan mempertahankan relasi yang ada dari himpunan data sebelumnya. Kebanyakan skema enkripsi homomorfik bekerja dengan baik pada data yang direpresentasikan sebagai bilangan bulat. Terdapat dua jenis operasi pada enkripsi homorfik, yaiut aditif dan multiplikatif. Selain itu, terdapat tiga jenis enkripsi homomorfik, yaitu:

1. *Partially homomorphic encryption* (PHE)

   PHE memungkinkan untuk dilakukan satu operasi tertentu untuk dilakukan pada *ciphertext*.

2. *Somewhat homomorphic encryption* (SHE)

   SHE memungkinkan untuk dilakukkannya dua operasi tertentu dengan jumlah operasi yang dapat dilakukan terbatas.

3. *Fully homomorphic encryption* (FHE)

   FHE memungkinan untuk dilakukkannya dua operasi tertentu pada *ciphertext*.

### E. Enkripsi Homomorfik

Enkripsi homomorfik adalah algoritma enkripsi yang memungkinan untuk dilakukannya komputasi terhadap data terenkripsi atau *ciphertext*. Secara matematis, enkripsi homomorfik melakukan transformasi dari suatu himpunan data menjadi himpunan data yang lain dengan mempertahankan relasi yang ada dari himpunan data sebelumnya. Kebanyakan skema enkripsi homomorfik bekerja dengan baik pada data yang direpresentasikan sebagai bilangan bulat. Terdapat tiga jenis enkripsi homomorfik, yaitu:

## III. Implementasi

Implementasi dari *paillier cryptosystem* menggunakan bahasa pemrogramman *python*. Berikut adalah kode dari implementasi operasi-operasi dasar dari *paillier*.

### A. Pembangkitan Kunci

Kode pembangkitan kunci ini seperti pada tabel 1. Dalam membangkitkan kunci, fungsi *generate_random_prime* digunakan untuk membangkitkan prima dengan panjang bit *bits_len*. Fungsi *random.randint* digunakan untuk membangkitkan sembarang bilangan bulat antara 1 dan $n^2 - 1$. Selain itu, fungsi *inverse_modulo* digunakan untuk menghitung nilai dari $a^{-1} \ (mod \ n)$.

Tabel 1. Kode pembangkitan kunci *paillier cryptosystem*

```
def generate_key_pair(bits_len):
    p = generate_random_prime(bits_len)
    q = generate_random_prime(bits_len)

    # public key
    n = p*q
    g = random.randint(1, n*n - 1)
    pub_key = (n, g)

    # private key
    a = (p-1) * (q-1)
    b = inverse_modulo(a, n)
    priv_key = (a, b)

    return priv_key, pub_key
```

### B. Enkripsi

Kode enkripsi seperti pada tabel 2. Pada Implementasi berikut, digunakan fungsi *random.randint* dan *pow*. Fungsi *pow* digunakan untuk menghitung perpangkatan dan dimodulo parameter ketiga.

Tabel 2. Kode enkripsi *paillier cryptosystem*

```
def encrypt(pub_key, m):
    n, g = pub_key
    n_sqr = n*n
    r = random.randint(1, n-1)
    c = pow(g, m, n_sqr) * pow(r, n, n_sqr) % n_sqr
    return c
```

### C. Dekripsi

Kode dekripsi seperti pada tabel 3. Pada implementasi berikut hanya digunakan fungsi *pow*.

Tabel 3. Kode dekripsi *paillier cryptosystem*

```
def L(x, n):
    return (x-1)/n

def decrypt(priv_key, n, c):
    a, b = priv_key
    n_sqr = n * n
    x = pow(c, a, n_sqr)
    m = (L(x, n) * b) % n
    return m
```

## IV. Aplikasi

Algoritma *paillier cryptosystem* memiliki sifat *partially homomorphic encryption* yang mendukung operasi adiktif. Sifat tersebut mendukung sistem *e-voting* yang aman. Selain itu, sifat tersebut juga mendukung sistem *e-money*.

## V. Kesimpulan

*Paillier cryptosystem* adalah algoritma kunci publik yang mudah diimplentasikan. Kunci publiik digunakan untuk Algortima ini memiliki sifat *partially homomorphic encryption*.

## REFERENCES

[1] Stallings, William, Cryptography and Network Security: Principles and Practice. Prentice Hall.

[2] Paillier, Pascal (1999). "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes". EUROCRYPT. Springer.