Implementasi Algoritma ElGamal dalam Enkripsi Gambar

Rayhan Fadhlan Azka - 13522095
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13522095@std.stei.itb.ac.id

***Abstract*—Tujuan dari makalah ini adalah untuk memperlihatkan tata cara mengimplementasikan algoritma ElGamal dalam enkripsi gambar. Salah satu permasalahan saat ini adalah rentannya pencurian data, baik itu berupa teks maupun gambar. Data ini dapat dicuri saat kita sedang mengirim data tersebut melalui jaringan yang tidak aman. Oleh karena itu dibuatlah suatu teknik enkripsi yang dimana data yang dikirim melalui jaringan tersebut adalah data asli yang telah diubah sedemikian rupa agar hanya orang yang dituju yang dapat membaca data tersebut. Misal saya tinggal di Indonesia dan teman saya tinggal di Spanyol. Jika saya ingin mengirim data ke teman saya secara rahasia, maka kami harus menyepakati semacam kunci yang dapat digunakan untuk mengkodekan pesan. Namun, saya tidak dapat menggunakan kunci yang sama terus-menerus karena hal itu dapat membuat data yang saya kirim rentan diretas. Disinilah Kriptografi ElGamal dapat menjadi solusi, kriptografi ini menggunakan dua jenis kunci, kunci privat dan kunci publik.**

***Keywords*—ElGamal, Enkripsi, Gambar, Kriptografi**

I. Pendahuluan

Perkembangan zaman telah membuat komunikasi menjadi jauh lebih mudah. Dalam era digital ini, pertukaran informasi menjadi semakin cepat dan luas. Bersamaan dengan itu, kebutuhan untuk melindungi privasi dan keamanan data semakin mendesak. Kriptografi, sebagai ilmu yang mempelajari tentang cara menyembunyikan informasi dari pihak yang tidak berhak, menjadi suatu hal yang penting dalam memastikan keamanan data yang dikirim dan disimpan.

Enkripsi gambar pada topik kali ini mungkin dilakukan karena setiap gambar memiliki pixel, dan setiap pixel memiliki komponen merah, hijau , dan biru yang bernilai bilangan bulat diantara 0 sampai 255. Masing-masing nilai warna tersebut nantinya akan di enkripsi dan menghasilkan gambar baru yang acak dan sangat berbeda dengan gambar awal. Lalu, dari gambar acak tersebut, hanya orang tertentu saja yang dapat mendekripsi gambar menjadi gambar semula.

Salah satu metode kriptografi yang akan digunakan kali ini adalah kriptografi ElGamal. Algoritma ElGamal dirancang khusus untuk skema kriptografi kunci publik, di mana kunci publik dan kunci privat yang berpasangan memungkinkan pengguna untuk melakukan enkripsi dengan kunci publik dan dekripsi dengan kunci privat.Algoritma ElGamal pada umumnya digunakan dalam enkripsi pesan teks, tapi hal ini tidak membuat enkripsi gambar menjadi mustahil. Makalah ini akan membahas penggunaan algoritma ElGamal dalam konteks kriptografi gambar, di mana gambar dianggap sebagai suatu bentuk data yang memerlukan perlindungan privasi.

II. Landasan Teori

A. Teori Bilangan

Teori bilangan adalah cabang matematika murni yang ditujukan untuk mempelajari bilangan bulat (integer) atau fungsi bernilai bilangan bulat. Bilangan bulat (integer) adalah bilangan yang tidak mempunyai pecahan desimal. Beberapa Teorema dan sifat-sifat yang terdapat pada Teori Bilangan :

1. Sifat Pembagian Bilangan Bulat

   Bilangan bulat tersebut memiliki suatu sifat, yaitu sifat pembagian pada bilangan bulat. Misalkan a dan b bilangan bulat, $a \neq 0$. a habis membagi b jika terdapat bilangan bulat c sedemikian sehingga $b = ac$. Sebagai contoh, $4 \mid 12$ karena $12/4 = 3$ (bilangan bulat) atau $12 = 4 \times 3$. Hal ini sesuai dengan

   $$a \mid b \text{ jika } b = ac, c \in \mathbb{Z} \text{ dan } a \neq 0$$

2. Teorema Euclidean 1

   Misalkan m dan n bilangan bulat, $n > 0$. Jika m dibagi dengan n maka hasil pembagiannya adalah q (quotient) dan sisanya r (remainder), sedemikian sehingga

   $$m = nq + r$$

   dengan $0 \leq r < n$. Artinya, ketika suatu bilangan bulat m dibagi oleh bilangan bulat positif n, kita dapat menemukan dua bilangan bulat, yaitu q dan r, sehingga m dapat diungkapkan sebagai hasil kali n dan q, ditambah dengan sisa r, di mana sisa r memenuhi $0 \leq r < n$. Sebagai contoh

   $$\frac{19}{7} = 2, \text{ sisa } 5$$
   $$19 = 7 \text{ x } 2 + 5$$

3. Pembagi Bersama Terbesar (PBB)

   Sifat lain pada bilangan bulat adalah pembagi bersama terbesar (PBB) atau dalam bahasa inggris greatest

common divisor (GCD). Misalkan a dan b bilangan bulat tidak nol, pembagi bersama terbesar (PBB) dari a dan b adalah bilangan bulat terbesar d sedemikian hingga

$$d \mid a \text{ dan } d \mid b$$

dalam hal ini kita nyatakan bahwa

$$\text{PBB}(a, b) = d$$

Sebagai contoh PBB(21,14) = 7 karena 7 merupakan bilangan bulat terbesar yang memenuhi 7 | 14 dan 7 | 21.

4. Teorema Euclidean 2

Misalkan m dan n bilangan bulat, dengan syarat n > 0 sedemikian sehingga $m = nq + r$, $0 \leq r < n$, maka

$$\text{PBB}(m, n) = \text{PBB}(n, r)$$

5. Algoritma Euclidean dalam Mencari PBB

PBB dari dua buah bilangan bulat dapat dicari dengan menggunakan algoritma Euclidean. Algoritma ini mengatakan jika ada dua buah bilangan bulat tak negatif m dan n, dengan $m \geq n$. Misalkan $r_0 = m$ dan $r_1 = n$. Lakukan secara berturut-turut pembagian untuk memperoleh

$r_0 = r_1 q_1 + r_2$, $0 \leq r_2 \leq r_1$
$r_0 = r_2 q_2 + r_3$, $0 \leq r_3 \leq r2$
.
.
.
$r_{n-2} = r_{n-1} q_{n-1} + r_n$, $0 \leq r_n \leq r_{n-1}$
$r_{n-1} = r_n q_n + 0$

Dari algoritma diatas, didapatkan

$\text{PBB}(m, n) = \text{PBB}(r_0, r_1) = \text{PBB}(r_1, r_2) =$
…
$= \text{PBB}(r_{n-2}, r_{n-1})$
$= \text{PBB}(r_{n-1}, r_n)$
$= \text{PBB}(r_n, 0)$
$= r_n$

Jadi, PBB dari m dan n adalah sisa terakhir yang tidak nol dari runtunan pembagian tersebut , sebagai contoh akan dicari PBB(35,14) , berdasarkan algoritma Euclidean

$35 = 14 \text{ x } 2 + 7$
$14 = 7 \text{ x } 2 + 0$

$PBB(35,14)$
$= PBB(14,7)$
$= PBB(7,0)$
$= 7$

6. Kombinasi Linier

Misalkan a dan b bilangan bulat positif, maka terdapat bilangan bulat m dan n sedemikian sehingga

$$\text{PBB}(a, b) = ma + nb$$

Dalam mencari m dan n, dapat digunakan algoritma euclidean dengan teknik penyulihan mundur. Sebagai contoh akan dicari m dan n dari PBB(17,7).

17 = 2 . 7 + 3 (i)
7 = 2 . 3 + 1 (ii)
3 = 3 . 1 + 0 (iii)

Susun (ii) menjadi:

1 = 7 - 2 . 3 (iv)

Susun (1) menjadi

3 = 17 – 2 . 7 (v)

Sulihkan (v) ke dalam (iv):

1 = 7 – 2 . (17 – 2 . 7) = 1 . 7 – 2 . 17 + 4 . 7
= 5 . 7 – 2 . 17

Atau

-2 . 17 + 5 . 7 = 1

Dari persamaan ini didapat PBB(17,7) = 1 dan nilai m = -2 dan nilai n = 5.

7. Relatif Prima

Dua buah bilangan bulat a dan b dikatakan relatif prima jika

$$\text{PBB}(a, b) = 1$$

Sebagai contoh 23 dan 7 relatif prima sebab PBB(23,7) = 1

8. Aritmatika Modulo

Misalkan a dan m bilangan bulat ($m > 0$). Operasi $a \bmod m$ memberikan sisa jika a dibagi dengan m. Hal ini dapat ditulis sebagai

$$a \bmod m = r$$

sedemikian sehingga

$$a = mq + r$$

dengan

$$0 \leq r < m$$

m disebut modulus, dan hasil aritmatika modulo m akan terletak pada rentang 0 dan $m - 1$ (inklusif). Beberapa contoh aritmatika modulo

- $23 \bmod 5 = 3$
- $27 \bmod 3 = 0$
- $6 \bmod 8 = 6$

9. Kongruensi

Jika terdapat a, b dan m yang memenuhi $a \bmod m = c$ dan $b \bmod m = c$, maka dapat dikatakan bahwa a kongruen dengan b dalam modulus m, atau ditulis : $a \equiv b \pmod m$ Untuk m > 0, maka juga berlaku : $m \mid (a - b)$. Sebagai contoh

$22 \bmod 5 = 2$
$17 \bmod 5 = 2$

Maka,

$$22 \equiv 17 \ (mod\ 5)$$

Dan berlaku,

5 | (22-17)

Jika $a \equiv b \ (mod\ m)$ dan c sembarang bilangan bulat, maka berlaku :

- $(a + c) \equiv (b + c) \ (mod\ m)$
- $a \ . \ c \equiv b \ . \ c \ (mod\ m)$
- $a^p \equiv b^p \ (mod\ m)$

Jika $a \equiv b \ (mod\ m)$ dan $c \equiv d \ (mod\ m)$, maka berlaku:

- $(a + c) \equiv (b + d) \ (mod\ m)$
- $a \ . \ c \equiv b \ . \ d \ (mod\ m)$

10. Invers Modulo

Di dalam aritmetika bilangan riil, balikan sebuah bilangan yang tidak nol adalah bentuk pecahannya sedemikian sehingga hasil perkalian keduanya sama dengan 1. Jika a adalah sebuah bilangan tidak-nol, maka balikannya adalah $\frac{1}{a}$ sedemikian sehingga $a \times \frac{1}{a} = 1$. Contoh: Balikan 4 adalah $\frac{1}{4}$, sebab $4 \times \frac{1}{4} = 1$.

Balikan modulo hanya bisa dicari jika persamaannya memenuhi syarat, yaitu untuk sebuah bilangan bulat a dan modulus m dengan $m > 1$ relatif prima atau $gcd(a, m) = 1$, maka balikan (invers) dari $a \ (mod\ m)$ ada.

Balikan dari $a \ (mod\ m)$ adalah bilangan bulat x sedemikian sehingga :

$$x \ . \ a \equiv 1 \ (mod\ m)$$

Dalam notasi lainnya,

$$a^{-1} \ (mod\ m) = x$$

11. Akar Primitif

Sebuah bilangan g disebut akar primitif dari n jika dan hanya jika untuk seluruh bilangan bulat a yang memenuhi $\text{PBB}(a, n) = 1$, terdapat sebuah bilangan bulat k yang memenuhi

$$g^k \equiv a \ (mod\ n)$$

Secara sederhana, g disebut akar primitif dari sebuah bilangan prima n jika perpangkatan

$a , a^2, \dots , a^{n-1}$ (dalam modulus n)

menghasilkan nilai yang berbeda beda. Sebagai contoh 3 adalah akar primitif dari 7 sebab

$$\begin{aligned}
3^1 &= 3^0 \times 3 \equiv 1 \times 3 = 3 \equiv 3 \pmod 7 \\
3^2 &= 3^1 \times 3 \equiv 3 \times 3 = 9 \equiv 2 \pmod 7 \\
3^3 &= 3^2 \times 3 \equiv 2 \times 3 = 6 \equiv 6 \pmod 7 \\
3^4 &= 3^3 \times 3 \equiv 6 \times 3 = 18 \equiv 4 \pmod 7 \\
3^5 &= 3^4 \times 3 \equiv 4 \times 3 = 12 \equiv 5 \pmod 7 \\
3^6 &= 3^5 \times 3 \equiv 5 \times 3 = 15 \equiv 1 \pmod 7
\end{aligned}$$

Disini, nilai $3^k \ mod\ n$ memiliki nilai yang berbeda-beda dari di antara k = 1 dan k = n -1.

12. Logaritma Diskrit

Jika g adalah akar primitif dari bilangan prima n, maka untuk bilangan bulat b kita dapat menemukan pangkat x sedemikian sehingga

$$b \equiv g^x \ (mod\ n) \ , \ 0 \leq x \leq (n - 1)$$

Pangkat x disebut logaritma diskrit dari b untuk basis $a \ (mod\ p)$. Dalam persoalan logaritma diskrit, diberikan $b \equiv a^x \ (mod\ p)$, carilah x yang memenuhi kekongruenan tersebut.

B. Kriptografi

Kriptografi adalah proses menyembunyikan atau mengkodekan informasi sehingga hanya orang yang dituju yang dapat membaca informasi tersebut.

Kriptografi berasal dari bahasa Yunani, crypto dan graphia. Crypto berarti secret (rahasia) dan graphia berarti writing (tulisan). Secara harafiah, kriptografi berarti suatu cabang ilmu pengetahuan yang mempelajari tentang cara mengamankan komunikasi dari pihak-pihak yang tidak diinginkan. Tujuan dari kriptografi adalah untuk menjamin pesan yang akan dikirimkan bersifat rahasia dan tidak dapat diidentifikasi oleh pihak yang tidak berhak.

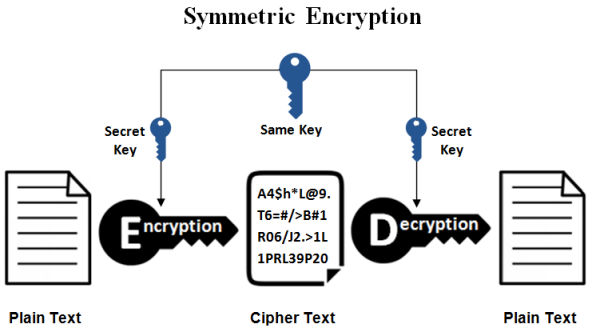
Kriptografi sendiri terdiri dari tiga komponen, yaitu

- Plainteks, yaitu pesan sebelum mengalami proses enkripsi.
- Cipherteks, yaitu pesan setelah proses enkripsi dan dapat di dekripsi.
- *Key*, yaitu formula atau kode yang dapat digunakan untuk mengubah pesan dari bentuk plainteks menuju cipherteks atau sebaliknya.
- Algoritma, algoritma pada kriptografi terbagi menjadi dua, yaitu algoritma enkripsi (mengubah plainteks menjadi cipherteks dengan key tertentu) dan algoritma dekripsi (mengubah cipherteks menjadi plainteks).

Kriptografi terbagi menjadi dua jenis, yaitu kriptografi simtetris (*symmetric cryptography*) dan kriptografi kunci-publik (*public-key cryptography*). Perbedaan antara kedua jenis kriptografi tersebut terletak pada penggunaan *key* saat melakukan enkripsi dan dekripsi.

Kriptografi simetris adalah kriptogradi dengan menggunakan satu *key* yang digunakan bersama oleh kedua pihak (pengirim data dan penerima data) untuk berbagi data yang terenkripsi. *Key* dalam jenis ini disebut simetris karena digunakan *key* yang sama untuk mengenkripsi dan mendekripsi data. Contoh sederhananya adalah pengirim mengenkripsi data menggunakan kata sandi, dan penerima harus mengetahui kata sandi tersebut untuk mengakses data.

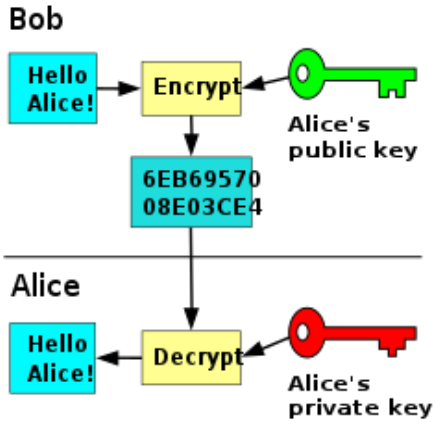
Beberapa contoh kriptografi simetris adalah Advanced Encryption Standard (AES), Blowfish, RC4 (Rivest Cipher 4), dan ChaCha20.



Gambar 2.1. Ilustrasi enkripsi simetris
Sumber : www.ssl2buy.com

Kriptografi kunci-publik adalah kriptografi dengan menggunakan dua jenis *key*, yaitu *public key* (kunci publik) dan *private key* (kunci privat). Kunci publik pada kriptografi ini dapat dikirim kepada siapapun dan kepemilikan terhadap kunci publik tidak cukup untuk dekripsi data. Lain halnya dengan kunci privat, kunci privat bersifat sangat rahasia dan kunci privat ini dapat digunakan untuk mendekripsi data.

Beberapa contoh kriptografi kunci publik adalah RSA (Rivest-Shamir-Adleman), Diffie-Hellman Key Exchange, Elliptic Curve Cryptography (ECC), Digital Signature Algorithm (DSA), dan ElGamal Encryption.



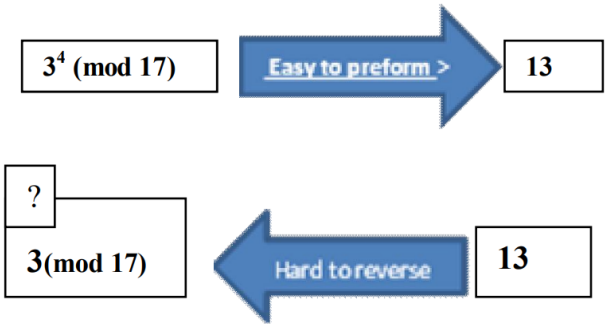
Gambar 2.2. Ilustrasi enkripsi kunci-publik
Sumber : www.wikipedia.com

C. Enkripsi ElGamal

Dalam kriptografi, sistem enkripsi ElGamal adalah algoritma enkripsi kunci asimetris dengan kriptografi kunci publik yang didasarkan pada *Diffie–Hellman key exchange*. Enkripsi ini dinamakan dari nama penemunya, Taher ElGamal, yang ditemukan pada tahun 1985. Algoritma ini banyak digunakan dalam *secure communication* pada jaringan yang tidak aman.

Tingkat keamanan pada algoritma enkripsi ElGamal terletak pada sulitnya menyelesaikan persoalan logaritma diskret.

$$b \equiv a^x \ (mod\ p)\ ,\ 0 \leq x \leq (p-1)$$



Gambar 2.3. Ilustrasi tingkat keamanan algoritma ElGamal
Sumber : Hasihim, 2014

Metode dari enkripsi ElGamal sendiri terdiri dari prosedur pembangkitan kunci, prosedur enkripsi, dan prosedur dekripsi. Misalkan Agent X ingin mengirim pesan ke Agent Y. Maka tahapannya pertama adalah Agent Y dapat menggunakan prosedur pembangkitan kunci dan menghasilkan kunci publik, tripel (y, g, p) dan kunci privat x.
Prosedur pembangkitan kunci yang dilakukan Agent Y adalah sebagai berikut

- Cari sembarang bilangan prima p (tidak rahasia)
- Pilih bilangan acak g, dengan syarat $g < p$, dan g adalah akar primitif dari p (tidak rahasia)
- Pilih bilangan acak x, dengan syarat $2 \leq x \leq p-2$ (rahasia, ini adalah kunci privat Agent Y)
- Hitung $y = g^x \ mod\ p$ (tidak rahasia)

Hasil dari algoritma ini menghasilkan

- Kunci Publik (y, g, p)
- Kunci Privat x

Lalu Agent X ingin mengirim pesan terenkripsi, misal pesan tersebut adalah m, Agent X dapat menggunakan Kunci Publik yang tersedia (y, g, p). maka prosedur enkripsi yang dapat dilakukan oleh Agent X adalah

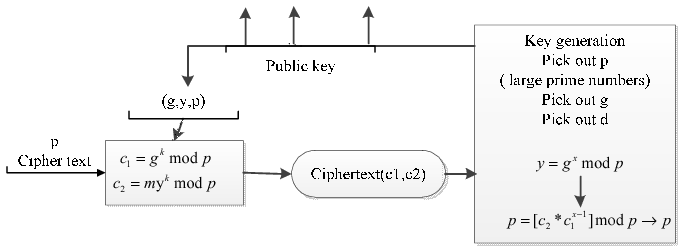
- Pilih bilangan acak k , dengan $1 \leq k \leq p-1$ (rahasia, ini adalah kunci privat Agent X)
- Hitung $c_1 = g^k \ mod\ p$ (tidak rahasia)
- Hitung $c_2 = y^k \ . \ m \ mod\ p$ (tidak rahassia)

Maka didapat pasangan cipherteks (c_1, c_2) yang nantinya dapat di dekripsi oleh Agent Y. Cipherteks (c_1, c_2) lalu dapat dikirim oleh Agent X ke Agent Y.

Lalu, ketika Agent Y ingin dekripsi cipherteks (c1,c2) menjadi plainteks m, Agent Y dapat melakukan dekripsi dengan prosedur

- Hitung $s = {c_1}^x \ mod\ p$
- Hitung s^{-1}, dimana ini adalah invers modulo dari s
- Dekripsi pesan cipherteks dengan menghitung $m = c_2 \ . \ s^{-1} \ mod\ p$

Dengan melakukan ketiga prosedur ini, pesan yang dikirim oleh Agent X menuju Agent Y dapat dikirim secara *secured* tanpa perlu memperlihatkan kunci privat kedua belah pihak.



Gambar 2.4. Ilustrasi algoritma ElGamal
Sumber : Zenggiang, 2014

D. Algoritma ElGamal dalam Enkripsi Gambar

Algoritma ElGamal umumnya digunakan untuk enkripsi teks, tetapi pada makalah ini penulis menggunakan algoritma ElGamal dalam melakukan enkripsi gambar. Hal ini dapat dilakukan karena pada setiap gambar, terdapat pixel seukuran panjang x lebar gambar, pixel-pixel itu sendiri direpresentasikan oleh nilai red, green, dan blue (RGB) yang masing masing bernilai diantara (0,255). Nilai RGB tersebut dapat diekstrak menjadi sebuah matriks, lalu kita dapat menerapkan algoritma ElGamal pada setiap elemen pada matriks tersebut. Prosedur untuk enkripsi gambar dengan algoritma ElGamal adalah

1. Baca masukan gambar dan konversi menjadi matriks RGB
2. Lakukan prosedur pembangkitan kunci, yaitu cari sembarang bilangan prima p, bilangan acak g (dengan $g < p$ dan g adalah akar primitif dari p), bilangan acak x ($2 \leq x \leq p-1$, ini adalah kunci privat), dan hitung $y = g^x \bmod p$
3. Lakukan prosedur enkripsi pada gambar, yaitu cari kunci privat k ($1 \leq k \leq p-1$), lalu hitung $c_1 = g^k \bmod p$
4. Lakukan traversal pada setiap pixel gambar, lalu lakukan prosedur enkripsi c_2 terhadap pixel-pixel tersebut dengan nilai plainteks m adalah elemen RGB masing-masing pixel (0,255), hitung $c_2 = y^k \cdot m \bmod p$, karena nilai c_2 dapat melebihi batas RGB (255), maka simpan nilai $c_2 \bmod 256$ pada matriks cipher image, matriks cipher image ini lalu akan di convert menjadi sebuah gambar yang telah terenkripsi.Karena nilai c_2 masih bersisa, simpan sisanya pada matriks c_2.
5. Lalu, untuk proses dekripsi, gabungkan matriks c_2 dan matriks cipher image yang didapat yang menghasilkan nilai c_2 original. Lalu hitung nilai $s = c_1^x \bmod p$
6. Lakukan traversal pada matriks c_2, dan hitung nilai plainteks $m = c_2 \cdot s^{-1} \bmod p$, nilai m inilah yang menjadi nilai plainteks atau nilai pixel-pixel RGB yang sesungguhnya.

III. IMPLEMENTASI

Pada makalah ini, penulis menggunakan bahasa pemrograman python untuk enkripsi gambar dengan algoritma ElGamal. Penggunaan bahasa python dalam algoritma ini disebabkan karena menurut penulis python memiliki library yang lengkap dan bahasanya lebih mudah dipahami dibanding bahasa pemrograman lain. Kekurangan dari python sendiri adalah waktu eksekusi yang lebih lama dibanding bahasa low-level lain dikarenakan lambatnya python dalam mengeksekusi *for loop*. Penulis memanfaatkan library numpy untuk pemrosesan matriks image yang telah diekstrak, library pillow untuk convert pemrosesan gambar, dan library random untuk meng-*generate* random number dalam key generator. Berikut adalah penjelasan beberapa fungsi yang digunakan dalam program ini

```python
def modularExp(a, b, n):
    result = 1
    count = 0
    while count < b:
        result = (result * a) % n
        count += 1
    return result
```

Gambar 3.1. Fungsi untuk menhitung a^b mod n
Sumber : arsip pengguna

```python
def findPrimitive(n):
    s = set()
    for i in range(2,n):
        for j in range(1,n):
            val = modularExp(i,j,n)
            s.add(val)
        if(len(s) == n-1):
            return i
        else:
            s.clear()
```

Gambar 3.2. Fungsi untuk mencari akar primitif dari bilangan prima n.
Sumber : arsip pengguna

```python
def gcdExtended(a, b):
    global x, y

    if (a == 0):
        x = 0
        y = 1
        return b

    gcd = gcdExtended(b % a, a)
    x1 = x
    y1 = y

    x = y1 - (b // a) * x1
    y = x1

    return gcd
```

Gambar 3.3. Fungsi untuk mencari PBB dan kombinasi linier
Sumber : arsip pengguna

Pada fungsi ini, ingin dicari nilai PBB(a,b) = xa + yb. Nilai PBB(a,b) dan x nantinya berguna untuk menghitung invers modulo.

```
def modInverse(a, p):

    g = gcdExtended(a, p)
    if (g != 1):
        return
        # tidak memiliki inverse
    else:
        res = (x % p + p) % p
        return res
```

Gambar 3.6. Fungsi untuk mencari inverse modulo
Sumber : arsip pengguna

Disini penulis mencari inverse dari $a \ mod \ p$ atau $a^{-1} \ mod \ p$. Invers modulo dari a dalam modulus p adalah suatu bilangan x sehingga $a \ . \ x \equiv 1 \ mod \ p$. Fungsi ini pertama-tama menghitung PBB dari a dan p dan koefisien x dan y sehingga ax + py = PBB(a, p). Jika a dan p relatif prima , maka a memiliki invers dalam modulus p, dan itu adalah $x \ mod \ p$, x didapat dari fungsi gcdExtended sebelumnya. Ekspresi (x % p + p) % p untuk memastikan agar nilai invers yang diambil positif.

```
def extractRGB(img_path):
    img = Image.open(img_path)
    img = img.convert('RGB')
    img_array = np.array(img)
    return img_array
```

Gambar 3.5. Fungsi untuk ekstrak fitur-fitur RGB pada gambar menjadi matriks
Sumber : arsip pengguna

Proses mengekstrak gambar menjadi matriks rgb, dengan nilai warna merah ada pada img_array[i,j,0], warna hijau ada pada img_array[i,j,1], dan warna biru pada img_array[i,j,2].

```
def save(img,img_path):
    img = Image.fromarray(img ,mode = 'RGB')
    img.save(img_path)
```

Gambar 3.6. Prosedur untuk konversi array menjadi image lalu di save di lokal
Sumber : arsip pengguna

```
def generateKey():
    p = generatePrime(256,1000)
    g = findPrimitive(p)
    x = random.randint(2,p-2)
    y = modularExp(g,x,p)

    return y,g,p,x
```

Gambar 3.7. Fungsi pembangkit kunci. Menghasilkan kunci publik (y,g,p) dan kunci privat x.
Sumber : arsip pengguna

Pada bagian ini, akan dicari nilai bilangan prima p, akar primitif dari p yaitu g, nilai y yaitu $g^x \ mod \ p$. Triple (y,g,p) akan menjadi kunci publik, dan x akan menjadi kunci privat.

```
def encryptC2(plainteks,y,k,p):
    c2 = modularExp(y,k,p) * plainteks % p
    return c2

def encryptC1(g,k,p):
    c1 = modularExp(g,k,p)
    return c1
```

Gambar 3.8. Fungsi enkripsi yang menghasilkan c1, dan c2.
Sumber : arsip pengguna

```
def decrypt(c1,c2,x,p):
    s = modularExp(c1,x,p)
    sInverse = modInverse(s,p)
    plainteks = (c2 * sInverse) % p
    return plainteks
```

Gambar 3.9. Fungsi dekripsi yang mengembalikan nilai plainteks dari nilai cipherteks
Sumber : arsip pengguna

```
def encryptImage(img_array,y,p,k,output_path):

    img_array = np.int64(img_array)
    c2_matrix = np.zeros_like(img_array)
    height, width, _ = img_array.shape

    for i in range(height):
        for j in range(width):
            pixel_data = img_array[i, j]
            r, g, b = pixel_data

            redC2 = encryptC2(r,y,k,p)
            greenC2 = encryptC2(g,y,k,p)
            blueC2 = encryptC2(b,y,k,p)

            img_array[i, j] = [redC2%256, greenC2%256, blueC2%256]

            c2_matrix[i,j] = [redC2 - (redC2%256) , greenC2 - (greenC2%256), blueC2 - (blueC2%256)]

    img_array = np.uint8(img_array)
    save(img_array,output_path)
    return c2_matrix
```

Gambar 3.10. Fungsi yang menenkripsi gambar menjadi gambar baru yang terenkripsi dan matriks c2.
Sumber : arsip pengguna

Pada fungsi ini, dilakukan pemrosesan pada matriks gambar plainteks. Cara kerjanya adalah dilakukan traversal untuk setiap pixel, lalu pada setiap pixel komponen merah , hijau, dan biru akan di enkripsi menjadi kunci c_2. Kunci c_2 yang dihasilkan disini nilainya tidak selalu berada pada rentang 0 dan 255, nilai c_2 ini bergantung pada nilai bilangan prima p yang digunakan jika bilangan prima yang digunakan sangat besar, maka nilai c_2 juga akan menjadi sangat besar, sesuai dengan $c_2 = y^k \ . \ m \ mod \ p$. Selanjutnya nilai c_2 akan disimpan di dua matriks, yaitu matriks img_array yang akan di convert menjadi gambar, dan matriks c2_matriks yang tetap disimpan sebagai matriks.

```python
def decryptImage(cipher_img,x,p,c1,c2_matrix):
    img_array = extractRGB(cipher_img)
    img_array = np.int64(img_array)
    height, width, _ = img_array.shape
    plainImg_matrix = np.zeros_like(img_array)

    for i in range(height):
        for j in range(width):
            pixel_data = img_array[i, j]
            r, g, b = pixel_data
            originalC2 = c2_matrix[i,j] + [r,g,b]

            plainImg_matrix[i,j,0] = decrypt(c1,originalC2[0],x,p)
            plainImg_matrix[i,j,1] = decrypt(c1,originalC2[1],x,p)
            plainImg_matrix[i,j,2] = decrypt(c1,originalC2[2],x,p)


    plainImg_matrix = np.uint8(plainImg_matrix)

    output_path = 'img/decryptedImg.jpg'
    save(plainImg_matrix,output_path)
```

Gambar 3.11. Fungsi yang mendekripsi sebuah gambar cipher
Sumber : arsip pengguna

Pada fungsi dekripsi gambar, akan di ekstrak gambar yang telah di enkripsi menjadi matriks. Lalu, akan dilakukan looping setiap pixel pada matriks, setiap elemen merah,hijau, dan biru pada matriks gambar ditambahkan dengan nilai pada elemen c2_matriks. Setelah ditambahkan, nilai-nilai tersebut dapat di dekripsi untuk mendapatkan nilai-nilai sesungguhnya pada masing-masing pixel.

```python
def elGamal(plain_img_path):
    img_array = extractRGB(plain_img_path)
    y,g,p,x = generateKey()

    k = random.randint(2,p-2)


    c1 = encryptC1(g,k,p)
    c2_matrix = encryptImage(img_array,y,p,k,"img/cipherImg.png")

    decryptImage("img/cipherImg.png",x,p,c1,c2_matrix)
```

Gambar 3.12. Algoritma utama
Sumber : arsip pengguna

IV. HASIL

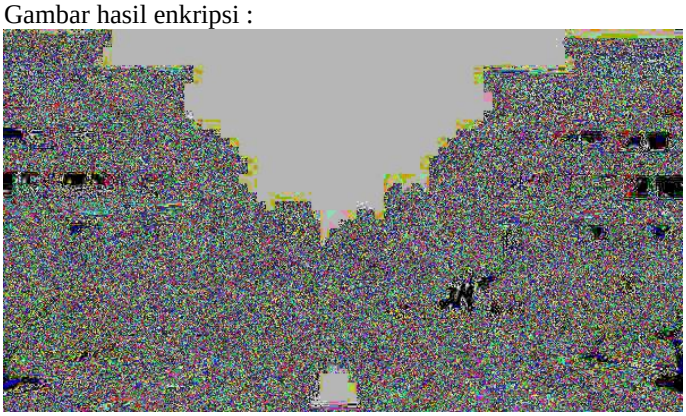
Berikut adalah beberapa hasil dari enkripsi dan dekripsi gambar menggunakan algoritma ElGamal.

Eksperimen 1

Gambar asli :

Gambar 4.1. gambar ITB
Sumber : detiknews.com

Gambar hasil enkripsi :



Gambar 4.2. Hasil enkripsi gambar ITB

Gambar hasil dekripsi :

Gambar 4.3 Hasil dekripsi gambar ITB yang telah di enkripsi
Eksperimen 2

Gambar asli :

Gambar 4.4 Gambar bunga
Sumber : Abcteach.com

Gambar hasil enkripsi :

Gambar 4.5. Hasil enkripsi gambar bunga

Gambar hasil dekripsi :

Gambar 4.6. Hasil dekripsi gambar bunga yang telah di enkripsi

V. KESIMPULAN

Algoritma ElGamal dapat mengenkripsi data berupa gambar menjadi gambar acak yang tidak ada maknanya, cipherteks lain berupa bilangan bulat, matriks dengan elemen berupa bilangan bulat. Setelah itu, hasil enkripsi tersebut dapat di dekripsi dengan menggabungkan cipherteks, kunci privat, dan kunci publik.

VI. UCAPAN TERIMA KASIH

Terima kasih kepada Tuhan Yang Maha Esa berkat rahmatnya penulis dapat menyelesaikan makalah ini. Tidak lupa juga ucapan terima kasih kepada dosen mata kuliah Matematika Diskrit, Dr. Nur Ulfa Maulidevi, S. T, M. Sc., Dr. Ir. Rinaldi Munir, M. T., dan Dr. Fariska Zakhralativa Ruskanda, S.T. yang telah membimbing penulis selama berkuliah di mata kuliah ini.

REFERENCES

[1] https://cp-algorithms.com/algebra/primitive-root.html. (diakses pada 10 Desember 2023)

[2] Daeri, A., Zerek,A., & Abuinjam, M. (2014). *ElGamal public-key encryption*, International Conference on Control, Engineering & Information Technology (CEIT'14). (diakses pada 10 Desember 2023)

[3] Raheem, H. (2014). *The Discrete Logarithm Problem in the ElGamal Cryptosystem over the Abelian Group U(n) Where n= pm ,or 2pm*. International Journal of Mathematics Trends and Technology (diakses pada 10 Desember 2023)

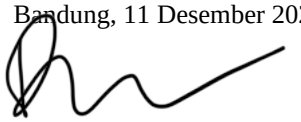
[4] https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended. (diakses pada 9 Desember 2023)

[5] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/14-Teori-Bilangan-Bagian1-2023.pdf (diakses pada 9 Desember 2023)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2023



Rayhan Fadhlan Azka - 13522095