Penerapan ECC untuk Enkripsi Pesan Berjangka Waktu

Dinah Kamilah Ulfa-13511087[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13510064@std.stei.itb.ac.id

***Abstraksi*— Pada makalah ini, akan dibahas implementasi metode kriptografi di mana pesan yang terenkripsi hanya dapat didekripsi dalam jangka waktu tertentu. Contohnya adalah pesan yang hanya dapat dibuka setelah tanggal yang ditentukan, pesan yang menjadi kadaluarsa dan tak dapat lagi diterjemahkan setelah melewati periode tertentu, atau pesan yang dapat dibuka pada jam-jam tertentu setiap harinya. Metode ini akan memanfaatkan algoritma *Eliptical Curve Cryptography* (ECC) dalam pengenkripsian pesan. Elemen waktu akan diintegrasikan dalam algoritma untuk mendapatkan fitur jangka waktu ini. Selanjutnya, sistem akan mengambil waktu dari clock komputer dalam mekanisme proses dekripsi, dan menggunakannya bersama dengan key dari pengguna untuk membuka pesan.**

Keywords—ECC, ElGamal, Time-Locked Encryption

I. PENDAHULUAN

Dalam pertukaran informasi, selalu terdapat ancaman berupa pencurian dan penggunaan data oleh pihak yang tidak berhak. Hal ini terutama beresiko terjadi pada pertukaran informasi melalui media digital, di mana perbuatan kriminal dapat dengan mudah lolos dari pengawasan. Oleh karena itu, penjagaan kerahasiaan informasi merupakan bagian penting dari sistem keamanan informasi digital.

Salah satu bidang ilmu keamanan informasi adalah kriptografi. Bidang ilmu kriptografi telah banyak menghasilkan algoritma yang melindungi keamanan informasi digital. Beberapa contoh dari algoritma tersebut adalah Advanced Encryption Standard (AES) dan algoritma kriptografi kunci publik seperti RSA, ElGamal, dan Diffie-Helman. Penerapan algoritma ini juga bervariasi, salah satunya adalah menggunakan ECC.

Algoritma ECC-ElGamal diakui sebagai algoritma dengan keaanan yang baik walaupun menggunakan kunci yang lebih pendek dari algoritma lain, seperti RSA. Algoritma ini bergantung pada kunci untuk enkripsi dan dekripsinya. Walaupun keamanan algoritma ini sudah cukup baik, keamanan tersebut masih dapat ditingkatkan dengan menambahkan faktor waktu pada algoritma enkripsi tersebut. Penambahan faktor waktu ini adalah dalam bentuk pembatasan waktu dimana pesan tersebut dapat didekripsi.

Oleh karena itu, makalah ini akan membahas metode pengenkripsian pesan dengan algoritma ECC di mana elemen waktu akan diintegrasikan dalam algoritma untuk mendapatkan fitur jangka waktu ini. Pengguna dapat mengatur tipe jangka waktu yang diberikan, seperti batas kadaluarsa atau batas waktu sebelum pesan dapat dibuka. Dalam dekripsinya, sistem akan mengambil waktu saat itu dari *clock* komputer dan menggunakannya bersama dengan key dari pengguna untuk membuka pesan.

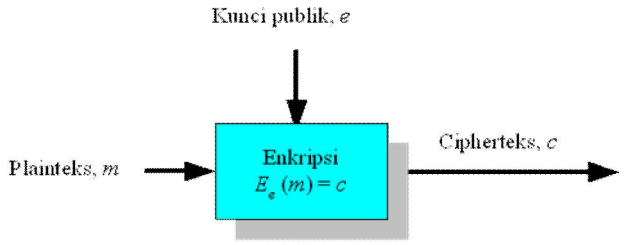
II. DASAR TEORI

A. Algoritma Kriptografi Kunci Publik

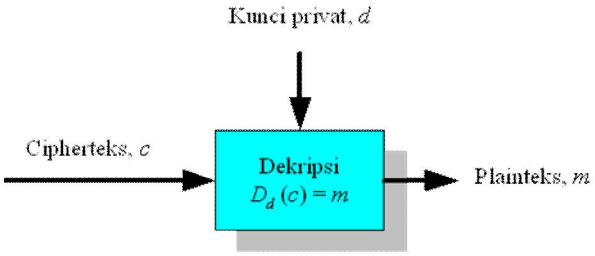
Kriptografi kunci publik pertama kali dibahas dalam makalah yang ditulis oleh Whitfield Diffie dan Martin Hellman. Pada kriptografi kunci publik, baik pengirim maupun penerima memiliki dua kunci, yaitu:

1. Kunci publik: untuk mengenkripsi pesan.
2. Kunci privat: untuk mendekripsi pesan.

Bila dituliskan dengan notasi aritmatik, maka fungsi enkripsi dan dekripsinya adalah $E_e(m) = c$ dan $D_d(c) = m$.



Gambar 1 Ilustrasi Enkripsi Kriptografi Kunci Publik



Gambar 2 Ilustrasi Dekripsi Kriptografi Kunci Publik

Adapun proses pengiriman pesan dengan algoritma ini adalah sebagai berikut.

- Misalkan: Pengirim pesan: Alice dan penerima pesan: Bob
- Alice mengenkripsi pesan dengan kunci publik Bob
- Bob mendekripsi pesan dengan kunci privatnya (kunci privat Bob)
- Sebaliknya, Bob mengenkripsi pesan dengan kunci publik Alice
- Alice mendekripsi pesan dengan kunci privatnya (kunci privat Alice)

Dengan mekanisme ini, kedua belah pihat dapat mengenkripsi dan dekripsi pesan tanpa harus berbagi kunci rahasia.

B. ElGamal

Algoritma ElGamal merupakan salah satu algoritma kriptografi kunci publik yang dirancang oleh Taher Elgamal. Keamanan algoritma ini terletak pada sulitnya menghitung logaritma diskrit yang diperlukan untuk menemukan kunci privat dari kunci publik.

Properti algoritma ElGamal:

1. Bilangan prima, p (tidak rahasia)
2. Bilangan acak, g ($g < p$) (tidak rahasia)
3. Bilangan acak, x ($x < p$) (rahasia, *key* privat)
4. $y = g^x \bmod p$ (tidak rahasia, *key* publik)
5. m (plainteks) (rahasia)
6. a dan b (cipherteks) (tidak rahasia)

Algoritma pembangkitan kunci publik dan privat ElGmal adalah sebagai berikut:

1. Pilih sembarang bilangan prima p (p dapat di-*share* di antara anggota kelompok)
2. Pilih dua buah bilangan acak, g dan x, dengan syarat $g < p$ dan $1 \le x \le p - 2$
3. Hitung $y = g^x \bmod p$.

   Hasil dari algoritma ini:

   - Kunci publik: tripel (y, g, p)

   - Kunci privat: pasangan (x, p)

Algoritma enkripsi ElGamal adalah sebagai berikut:

1. Susun plainteks menjadi blok-blok m_1, m_2, …, (nilai setiap blok di dalam selang $[0, p - 1]$.
2. Pilih bilangan acak k, yang dalam hal ini $1 \le k \le p - 2$.
3. Setiap blok m dienkripsi dengan rumus

$$a = g^k \bmod p$$
$$b = y^k m \bmod p$$

Pasangan a dan b adalah cipherteks untuk blok pesan m. Jadi, ukuran cipherteks dua kali ukuran plainteksnya.

Algoritma Dekripsi ElGamal:

1. Gunakan kunci privat x untuk menghitung

   $(a^x)^{-1} = a^{p-1-x} \bmod p$
2. Hitung plainteks m dengan persamaan:

$$m = b/a^x \bmod p \quad = b(a^x)^{-1} \bmod p$$

C. Eliptic Curve Cryptography

Eliptic Curve Cryptography merupakan sistem kriptografi kunci publik yang memanfaatkan persamaan kurva eliptik. Algoritma ini dirancang oleh diajukan oleh Neal Koblitz dan Victor S. Miller. Kurva eliptik merupakan kurva dengan bentuk umum persamaan:

$$y^2 = x^3 + ax + b$$

dengan syarat $4a^3 + 27b^2 \neq 0$. Tiap nilai a dan b berbeda memberikan kurva eliptik yang berbeda.

Penggunaan kurva eliptik dalam ECC adalah sebagai berikut:

Dua pihak yang berkomunikasi menyepakati parameter data sebagai berikut:

1. Nilai a, b, dan bilangan prima p dari persamaan kurva eliptik $y^2 = x^3 + ax + b \bmod p$

2. Grup eliptik yang dihitung dari persamaan kurva eliptik

3. Titik basis (base point) B (X_B , Y_B) , dipilih dari grup eliptik untuk operasi kriptografi.

Setiap pengguna membangkitkan pasangan kunci publik dan kunci privat

1. Kunci privat = integer x, dipilih dari selang [1, p – 1]
2. Kunci publik = titik Q, adalah hasil kali antara x dan titik basis B: Q = x· B

Dalam pengenkripsiannya, pertama-tama plainteks M dikode menjadisebuah titik, P_M dari kurva eliptik.Setelah itu, pengirim memilih bilangan acak lain, k, dari selang [1, p-1]. Cipherteks adalah pasangan titik

$$P_C = [\ (kB), (P_M + kP_B)\ ]$$

Dengan P_B merupakan kunci publik penerima.

Dalam proses dekripsi, penerima mula-mula menghitung hasil kali titik pertama P_C dengan kunci privatnya, b.

$$b \cdot (kB)$$

Selanjutnya, penerima kemudian mengurangkan titik kedua dari P C dengan hasil kali di atas

$$(P_M + kP_B) - [b \cdot (kB)] = P_M + k \cdot (bB) - b \cdot (kB) = P_M$$

Setelah P_M didapatkan, penerima mendecode P_M kembali ke plainteks untuk membaca pesan.

III. IMPLEMENTASI SOLUSI

Implementasi ECC untuk pesan berjangka waktu ini dibuat dalam bahasa Java. Dalam pengenkripsian dan pendekripsiannya, metode ini memiliki dua tahap. Yang pertama adalah pengenkripsian menggunakan ECC yang diintegrasikan dengan elemen waktu. Selanjutnya, informasi waktu ini akan dienkripsi lagi menggunakan algoritma ECC biasa. Kunci privat akan digunakan untuk mendekripsi informasi waktu ini. Bila waktu yang didapat dari *clock* komputer sesuai dengan batas waktu yang dimiliki ciphertext, maka pesan akan terbuka.

A. *ECC Enkripsi dengan Elemen Waktu.*

Seperti yang telah dijelaskan dalam dasar teori, parameter *elliptic curve* adalah p,a,b, dan G, dengan G adalah poin basis. Dalam ECC biasa, seluruh parameter ini telah disepakati sebelumnya oleh kedua belah pihak. Untuk implementasi ini, nilai a, b, dan G ditentukan oleh parameter batas waktu yang diberikan pengirim pesan. Adapun implementasi fungsi adalah sebagai berikut:

```
public ECC(long start, long finish){
    a = new
BigInteger(Long.toString(start));
    b = new
BigInteger(Long.toString(finish));
    p = new
BigInteger("627710173538668076383578942320766641608390870039032496127
9");
    n = new
BigInteger("627710173538668076383578942317605901376719477318284228408
1");
    k = 20; int m = 11;
    G = new Point(new
BigInteger("188da80eb03090f67cbf20eb43a18800f4ff0afd82ff1012",16),
                  new
BigInteger("07192b95ffc8da78631011ed6b24cdd573f977a11e794811",16));
    Gt = getBasePoint(a, b, m);
}
```

Dengan `start` adalah batas waktu awal dalam milidetik, dan `finish` adalah batas waktu akhir dalam milidetik. Dari nilai `start` dan `finish` ini, dihitung poin Gt dengan fungsi:

```
public Point getBasePoint(BigInteger a,
BigInteger b, int M){
        BigInteger x=BigInteger.ZERO,
y=BigInteger.ZERO;
        boolean Found=true;
        for(int i=1;i<=k-1;i++){
            x =
BigInteger.valueOf((M*k+i));
            BigInteger ry2 =
((x.pow(3)).add(a.multiply(x))).add(b);
            try{
                y =
IntegerFunctions.ressol(ry2,p);
            }catch(Exception e){
                Found=false;
            }
            if (Found) break;
        }
        return new Point(x,y);
    }
```

Setelah itu, pesan akan dienkripsi menggunakan poin basis yang didapat dari pertambahan antara poin G *time* dan poin G dari tahap selanjutnya yang menggunakan ECC biasa.

```
private void EncryptT() {
    for(int i=0;i<PlainBytes.length;i++){
        Point Pm =
EncodeMessage(PlainBytes[i]);
        BigInteger Km = GenerateBig(192);
        Point C1 = ScalarMult(G,Km);
        Point C2 =
AddPoint(Pm,AddPoint(Gt,ScalarMult(PubKey
,Km)));
        PointCipher.add(new
ECCipher(C1,C2));
}
```

Selanjutnya, waktu start dan finish dienkripsikan menggunakan poin G biasa sebagai basisnya. Hasil enkripsi ini kemudian digabungkan di bagian depan dengan pesan terenkripsi sebelumnya.

B. *ECC Dekripsi dengan Elemen Waktu.*

Pertama-tama, informasi mengenai jangka waktu pada cipherteks akan didekripsi menggunakan algoritma ECC biasa dengan menggunakan kunci privat dan titik basis G yang telah disepakati. Selanjutnya akan diambil waktu pada komputer saat itu menggunakan fungsi Java *System.currentTimeMillis()*. Lalu dicek apakah waktu saat itu berada dalam jangka waktu yang ditentukan. Bila tidak, sistem akan mengeluarkan error. Bila ya, maka sistem akan menggenerate poin basis Gt menggunakan fungsi getBasePoint().

```
private void Decrypt() {
    long timeNow =
System.currentTimeMillis();
    byte[] TempPlain = new byte[4];
    for(int i=0;i<8;i++){
        Point Pm =
AddPoint((PointCipher.get(i).C2),NegatePo
int(ScalarMult((PointCipher.get(i).C1),Pr
ivKey)));
```

```
        TempPlain[i] =
((Pm.x.subtract(BigInteger.ONE)).divide(B
igInteger.valueOf(k))).byteValue();
        if (i==3) {
           date1 = byteToLong (TempPlain);
           TempPlain = new byte[4];
        }
    }
    date2 = byteToLong (TempPlain);

    if (timeNow < Math.max(millis1,
millis2) && timeNow > Math.min(millis1,
millis2)) {
        Gt = getBasePoint(date1, date2,
m);
        DecryptT();
    } else {
        System.out.println("Time doesn't
match");
    }
}
```

Selanjutnya poin basis Gt akan digunakan bersamaan dengan kunci privat dan poin basis G untuk mendekripsi pesan.

```
private void DecryptT() {
    byte[] TempPlain = new
byte[PointCipher.size()];
    for(int
i=8;i<PointCipher.size();i++){
        Point Pm =
AddPoint((PointCipher.get(i).C2),NegatePo
int(AddPoint(Gt,ScalarMult((PointCipher.g
et(i).C1),PrivKey))));
        TempPlain[i] =
((Pm.x.subtract(BigInteger.ONE)).divide(B
igInteger.valueOf(k))).byteValue();
    }

    PlainBytes = TempPlain;
}
```

IV. Pengujian

Parameter:

Plainteks: “Kriptografi IF 2015”

Batas waktu awal: 2015-05-11 10:00:00

Batas waktu akhir: 2015-05-11 11:00:00

Waktu pembukaan: 2015-05-11 12:00:00

Hasil cipherteks

f2b0259b88441ad643d05d85c60e01bfbe3e000658b76a3339303
5567220236946818256635941211400482046850057304525444
56,2213520819121794381410290294369712279034492015532
990376728197966767842356121229719770804003042563823
0897235554617585.....

Hasil dekripsi

Time doesn't match

Dengan parameter:

Batas waktu awal: 2015-05-11 12:00:00

Batas waktu akhir: 2015-05-11 13:00:00

Waktu pembukaan: 2015-05-11 12:30:00

Hasil cipherteks

b04c4b57314cf90ac98e7dfa5dfc599781b0bb61e953966c2381
3899680875080805684574424963964660445578007217596 2
6997,6001374270597163159213943156506859211604432 37
3538260259094591232752367917697437246122587783388 0
3356951229913055534.....

Hasil dekripsi

Kriptografi IF 2015

V. Kesimpulan

Berdasarkan pembahasan di makalah ini, dapat disimpulkan bahwa ECC dapat digunakan untuk mengenkripsi pesan berjangka waktu dengan metode yang mirip dengan *timestamp*. Cipherteks ditandai dengan waktu pembukaan pesan, untuk selanjutnya dicocokkan dengan waktu pada komputer penerima untuk validasi. Diharapkan kedepannya dapat ditemukan algoritma yang lebih aman untuk menyembunyikan waktu pembukaan, sehingga jangka waktu tidak perlu diterakan pada pesan untuk proses dekripsi.

Daftar Pustaka

[1] Padma Bh, D.Chandravathi, P.Prapoorna Roja, “Encoding and Decoding of a message in the Implementation of Elliptic Curve Cryptography uing Koblitz Method”, *International Journal on Com puter Science and Engineering* , pp. 1904-1907, 2010.

[2] Munir, Rinaldi, Kriptografi Kunci Publik, Program Studi Teknik Informatika.
[3] Munir, Rinaldi, Algoritma RSA, Program Studi Teknik Informatika.
[4] Munir, Rinaldi, Algoritma ElGamal, Program Studi Teknik Informatika.

Bandung , 11 Mei 2015

Dinah Kamilah Ulfa/13511087

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.