Studi dan Analisis Perbandingan Antara Algoritma El Gamal dan Cramer-Shoup Cryptosystem

Yudhistira 13508105[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*if18105@students.if.itb.ac.id*

***Abstrak*—Algoritma El Gamal merupakan algoritma kunci publik yang masih terbilang baru. Algoritma ini terbilang aman dan banyak digunakan untuk tanda tangan digital. Beberapa tahun setelah kemunculan El Gamal, muncul algoritma baru yang bernama Cramer-Shoup. Dikatakan bahwa Cramer-Shoup ini merupakan pengembangan dari Algoritma El Gamal. Dalam makalah ini, dilakukan analisis dan perbandingan antara kedua algoritma.**

***Kata kunci*—El Gamal, Cramer, Shoup, Cryptosystem**

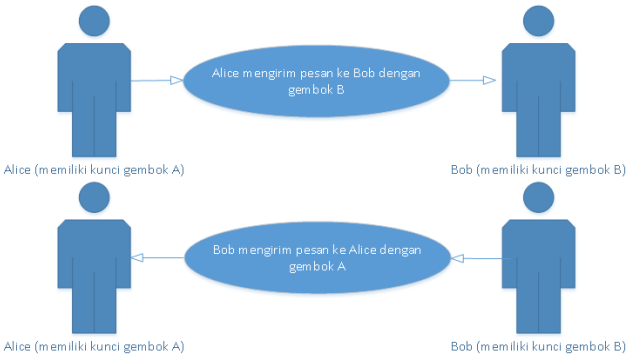
I. INTRODUCTION

Algoritma kunci publik merupakan salah satu algoritma modern yang banyak dipakai untuk berbagai macam keperluan. Ada banyak varian algoritma kunci publik. Orang banyak menyukai algoritma kunci publik karena algoritma ini sangat sulit untuk dipecahkan. Diperlukan perhitungan yang sangat kompleks untuk memecahkan algoritma ini.

Algoritma kunci publik sering kali menjadi alternatif algoritma penandatanganan dokumen dan untuk enkripsi. Konsep kunci publik ini pertama kali dikemukakan oleh 2 orang ilmuwan asal *University of Stanford*, yang bernama Whitfield Diffie dan Martin Hellman, dalam buku yang berjudul *New Direction in Cryptography.*

Konsep dasar algoritma kunci publik sebetulnya sangat sederhana. Asumsikan Alice ingin mengirim pesan ke Bob. Alice memasukan pesan dan gembok milik Alice sendiri ke dalam sebuah kotak, lalu mengunci kotak tersebut dengan gembok milik Bob. Lalu saat pesan sampai ke Bob, Bob menggunakan kunci milik Bob sendiri untuk membuka gembok kotak tersebut. Untuk membalas pesan, Bob menaruh pesan dan gembok milik Bob di dalam kotak, lalu mengunci kotak dengan gembok milik Alice. Skema dari pertukaran pesan ini dapat dilihat di gambar 1.

Beberapa algoritma kunci publik yang populer digunakan sebagai alat untuk menandatangani dokumen adalah Diffie-Helman, RSA, dan ElGamal. Dari ketiga algoritma tersebut, algoritma ElGamal merupakan algoritma yang paling baru dan paling mutakhir.



Gambar 1: Skema Algoritma kunci publik

Namun demikian, perkembangan algoritma ElGamal tidak berhenti sampai di situ. Hampir 2 dekade setelah kemunculan ElGamal, ditemukan sebuah metode enkripsi kunci publik baru yang diberi nama Cramer-Shoup *Cryptosystem.*

II. ALGORITMA EL GAMAL

Algoritma El Gamal adalah salah satu varian dari algoritma kunci publik yang ditemukan pada tahun 1985 oleh seorang ilmuwan bernama Taher Elgamal. Algoritma ini pertama kali diterbitkan di dalam sebuah makalah berjudul *"A public key cryptosystem and a signature scheme based on discrete logarithms".*

Sesuai dengan judul makalhnya, kekuatan dari algoritma ini diciptakan dari sulitnya menghitung logaritma diskrit. ElGamal mendasarkan algoritmanya kepada sbuah asumsi bahwa logaritma diskrit tidak dapat ditemukan pada waktu yang tepat, dan inverse operasi perpangkatan dapat dihitung dengan efisien.

Ada 3 tahap yang dibutuhkan untuk menggunakan algoritma ElGamal. Tahap pertama adalah tahap *Key Generation*. Keluaran dari tahap ini adalah pasangan kunci yang terdiri atas kunci publik dan kunci privat.

Untuk membangkitkan kunci, dibutuhkan sebuah bilangan prima acak *p*. Lalu pilih bilangan acak *g* dengan $g < p$, dan bilangan acak *x* dengan $1 \leq x \leq p-2$. Setelah itu, hitung $y = g^x \bmod p$. Hasil dari algoritma pembangkitan kunci ini adalah pasangan kunci privat (x,p) dan kunci publik (p,g,y).

Tahap kedua dari algoritma ini adalah tahap enkripsi. Dalam enkripsi, yang dibutuhkan adalah *plaintext*, sebuah bilangan acak *k*, dan kunci publik.

Langkah pertama enkripsi adalah dengan membagi plaintext menjadi blok blok plaintext. Nilai dalam tiap blok boleh antara 0 sampai *p-1*. Setelah itu, dipilih bilangan acak *k* dengan $1 \leq k \leq p-2$. Setelah itu, setiap blok plaintext dienkripsi dengan formula:

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

Pasangan *a* dan *b* merupakan ciphertext untuk sebuah blok plaintext. Panjang dari *ciphertext* adalah 2 kali dari panjang *plaintext*.

Tahap ketiga dari algoritma ini adalah tahap dekripsi. Dalam tahap ini, yang diperlukan adalah *ciphertext* dan kunci privat.

Langkah pertama dari algoritma ini adalah menghitung nilai *x* dari persamaan berikut:

$$(a^x)^{-1} = a^{p-1-x} \bmod p$$

Setelah itu, dicari plaintextnya (*m*) dengan persaman sebagai berikut:

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

Variabel m dihitung untuk semua blok ciphertext yang ada, kemudian digabungkan. Maka pesan awal akan tertampil.

III. CRAMER-SHOUP CRYPTOSYSTEM

Cramer-Shoup *Cryptosystem* merupakan algoritma kunci simetri yang ditemukan oleh 2 orang ilmuwan, yaitu Ronald Cramer dan Victor Shoup pada tahun 1998. Algoritma ini dibuat sebagai ekstensi dari algoritma ElGamal. Cramer-Shoup memiliki kelebihan kekebalan terhadap serangan ACCA (*Adaptive Chosen-Ciphertext Attack)*.

ACCA mengasumsikan bahwa pihak penyerang memiliki akses terhadap dekripsi oracle yang akan mendeskripsikan *ciphertext* manapun menggunakan skema kunci dekripsi rahasia. *Adaptive* memiliki arti bahwa penyerang memiliki akses terhadap dekripsi oracle baik sebelum maupun sesudah penyerang mengobservasi target ciphertext yang spesifik untuk dikenakan serangan.

Sebetulnya, Cramer-Shoup bukan skema pertama yang diajukan untuk menyediakan keamanan terhadap serangan ACCA. Rackoff-Simon, Folev-Dwork-Naor juga mengajukan skema untuk menyediakan keamanan terhadap serangan CCA. Namun, skema-skema tersebut tidak efisien dalam biaya komputasi.

Langkah-langkah untuk melakukan proses ACCA adalah sebagai berikut:

Pertama, pembangkitan kunci dijalankan dengan input berupa parameter keamanan. Kemudian, pihak penyerang membuat query yang bervariasi pada dekripsi oracle untuk mendekripsi ciphertext yang telah dipilih.

Kemudian, penyerang memilih 2 pesan. Lalu, kedua pesan dikirimkan pada enkripsi oracle. Enkripsi oracle kemudian memilih bit b secara random kemudian mengenkripsi pesan. Ciphertext yang cocok kemudian dikirim ke penyerang. Setelah penyerang menerima ciphertext, penyerang melanjutkan dengan melakukan query pada dekripsi oracle, dengan output yang dihasilkan berbeda dengan enkripsi oracle. Dan terakhir, penyerang mengeluarkan b sebagai tebakan untuk nilai b.

Cramer-Shoup telah dinyatakan sebagai sebuah algoritma dengan skema yang efektif, yang terbukti tahan terhadap serangan ACCA.

Seperti halnya algoritma ElGamal, Cramer-Shoup memiliki 3 tahap dalam penggunaan algoritma. Yaitu tahap pembangkian kunci, enkripsi, dan dekripsi.

Pada tahap pembangkitan kunci, proses yang terjadi adalah sebagai berikut:

1. Dibangkitkan g_1, g_2 di mana kedua bilangan itu adalah himpunan bagian dari G. G adalah kelompok bilangan prima q yang berukuran sangat besar.
2. Dipilih 6 bilangan acak (x_1 , x_2 , y_1 , y_2 , z) dengan rentang 0 sampai *q-1*.
3. Kemudian dihitung variabel c,d, dan h dengan rumus sebagai berikut:

   $c = g_1^{x1} g_2^{x2}$
   $d = g_1^{y1} g_2^{y2}$
   $h = g_1^{z}$

4. Dari perhitungan di atas, didapat kunci publik(g_1, g_2, c, d, h, H) dan kunci privat (x_1, x_2, y_1, y_2, z). *H* adalah fungsi hash satu arah.

Pada tahap enkripsi, yang dilakukan adalah sebagai berikut:

1. Pesan m dikonversi menjadi elemen G.
2. Dipilih bilangan acak r dalam rentang 0 sampai *q-1*. Lalu dihitung nilai u_1, u_2, e, a, dan v dengan menggunakan kunci publik. Berikut adalah formulanya:

   $U_1 = g_1^x$
   $U_2 = g_2^x$
   $e = h^x m$
   $a = H(u_1, u_2, e)$
   Dengan H merupakan fungsi Hash.
   $v = c^x d^{ra}$
3. Didapat ciphertext (u_1,u_2,e,v)

Pada tahap dekripsi, dilakukan dekripsi dengan menggunakan kunci privat. Prosesnya adalah sebagai berikut:

1. Nilai H dihitung dengan menggunakan fungsi Hash

$a = H(u_1, u_2, e)$

2. Jika hasil $u_1^{x1+y1a} \cdot u_2^{x2+y2a} = v$, maka $m = e/u_1^z$. Akan tetapi, jika hasil tidak sama dengan v, maka proses akan berhenti dam tidak menampilkan pesan.

IV. PERBANDINGAN ANTARA ALGORITMA ELGAMAL DAN CRAMER-SHOUP

Algoritma ElGamal memiliki performansi yang sangat baik. Begitu juga dengan Cramer-Shoup. Kedua algoritma secara umum memiliki kelebihan dan kekurangan masing-masing.

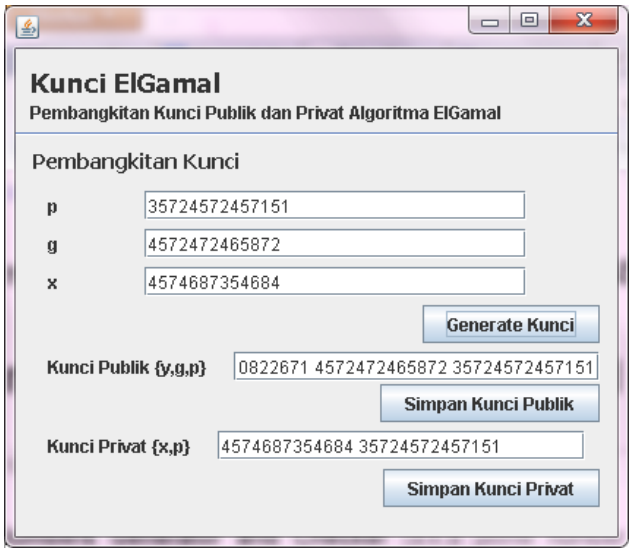
Algoritma ElGamal relatif lebih mudah diimplementasikan. Langkah-langkahnya berdasarkan matematika modulo yang tidak terlalu sulit. Sedangkan Cramer-Shoup relatif lebih sulit diimplementasikan.

Namun demikian, Cramer-Shoup memiliki keunggulan bila dibandingkan dengan ElGamal. Cramer-Shoup memiliki kekebalan terhadap serangan ACCA (*Adaptive Chosen-Ciphertext Attack*). Serangan ini merupakan serangan yang dirancang dengan cara menyadap dekriptor oracle. Lain halnya dengan Cramer Shoup, ElGamal tidak memiliki kekebalan terhadap serangan ini.

Perbedaan lainnya, Cramer Shoup sedikit lebih kompleks dari ElGamal. Kunci publik El Gamal hanya terdiri atas 3 variabel. Sedangkan kunci publik Cramer-Shoup terdiri atas 6 karakter.Begitu juga dengan kunci privatnya. Kunci privat elgamal terdiri atas 2 variabel. Sedangkan kunci privat Cramer-Shoup terdiri atas 5 variabel.

Untuk menangkal serangan ACCA, Cramer-Shoup memiliki fungsi Hash yang dimasukkan ke dalam algoritmanya. Algoritma ElGamal tidak memiliki fungsi Hash, sehingga lebih rawan terhadap serangan ACCA yang banyak digunakan oleh para penyerang.

Berikut adalah hasil implementasi dari algoritma ElGamal:



Didapat hasil sebagai berikut:

p = 35724572457151

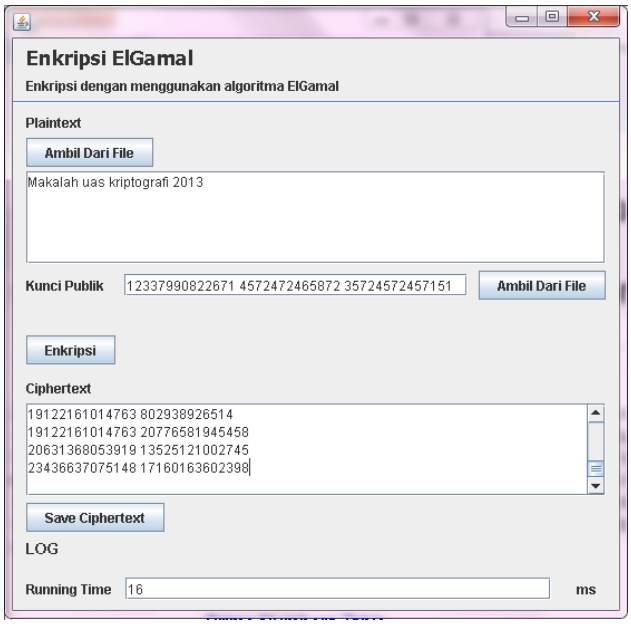
g = 4572472465872

y = 4574687354684

Kunci Publik = 12337990822671 4572472465872 35724572457151

Kunci Privat = 4574687354684 35724572457151

Kemudian pesan dienkripsi. Didapat hasil sebagai berikut:



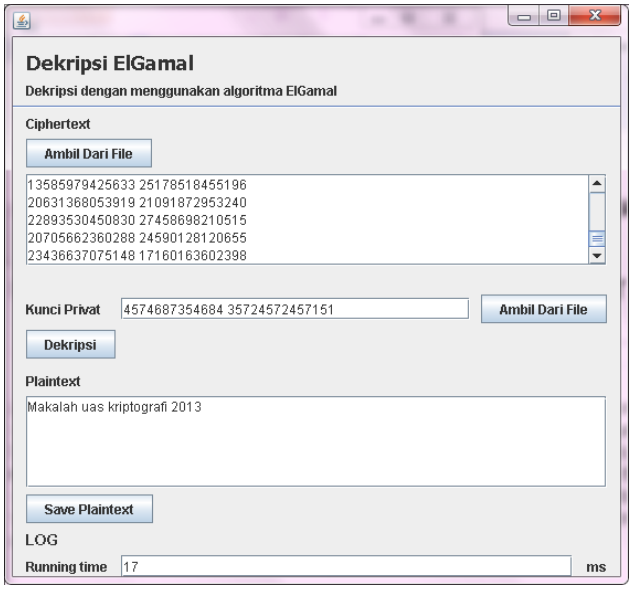
Pesan : Makalah uas kriptografi 2013

Ciphertext:

```
1 77
22893530450830 21997665955170
19122161014763 3147272201026
20631368053919 19483490054995
13585979425633 31390641100560
31261157843733 3936995880860
20705662360288 4801553363945
1 32
19122161014763 10452774477759
22893530450830 21997665955170
1001029747896 20061316903923
4572472465872 1845409296811
20631368053919 23701864645643
1 114
31261157843733 14205649730241
13585979425633 16675669678884
20631368053919 20353487285796
31261157843733 12975997002989
20631368053919 29159429300814
19122161014763 4688666549024
31261157843733 3936995880860
22893530450830 297303454474
29902336929973 21696945089593
13585979425633 25178518455196
20631368053919 21091872953240
```

```
22893530450830 27458698210515
20705662360288 24590128120655
23436637075148 17160163602398
```

Setelah itu, dilakukan dekripsi terhadap ciphertext dengan menggunakan kunci privat. Hasilnya adalah sebagai berikut:



Dan pesan pun kembali seperti semula.

Berikut adalah algoritma enkripsi yang digunakan dalam program di atas:

```
//Enkripsi
        Util util = new Util();
        ElGamal gamal = new ElGamal();

        String temp = "";
        String temp2 = "";
        String temp3 = "";

        String kpublik = jTextField1.getText();
        String plaintext = jTextArea1.getText();
        String ciphertext = "";

        String[] kuncipublik = kpublik.split(" ") ;
        String temp_y = kuncipublik[0];
        String temp_g = kuncipublik[1];
        String temp_p_a = kuncipublik[2];



        BigInteger y = new BigInteger(temp_y);
        BigInteger g = new BigInteger(temp_g);
        BigInteger p_a = new BigInteger(temp_p_a);

        BigInteger p = new BigInteger("1");

        System.out.println(plaintext);
        Character[] charArray = util.toCharacterArray(plaintext);

        int[] asciiArray = util.toAsciiArray(charArray);


        ArrayList<BigInteger> arraystring = gamal.convertArrayAscii(asciiArray);

        String[][] hasil = gamal.Enkrip(p_a, g, y, arraystring);

           int j,k;
           k=0;
           for(j=0;j<hasil.length;j++)
           {
              ciphertext=ciphertext+hasil[j][0]+" ";
              ciphertext=ciphertext+hasil[j][1]+"\n";
           }

        jTextArea2.setText(ciphertext);
```

Dan ini adalah algoritma dekripsinya:

```
        Util util = new Util();
        ElGamal gamal = new ElGamal();

        String temp = "";
        String temp2 = "";
        String temp3 = "";

        String kprivat = jTextField2.getText();
        String ciphertext = jTextArea1.getText();
        String plaintext = "";

        String[] kunciprivat = kprivat.split(" ");
        String temp_x = kunciprivat[0];
        String temp_p_b = kunciprivat[1];

        BigInteger x = new BigInteger(temp_x);
        BigInteger p = new BigInteger(temp_p_b);

        String[] temporary = ciphertext.split(" ");
        for (int i = 0; i < temporary.length; i++) {
           System.out.println("ini " + temporary[i]);

        }
        String[][] tobedecrypted = new String[temporary.length/2][2];
        int i=0;
        int j=0;
        System.out.println("kegilaan ini panjangnya segini "+temporary.length);
        while(i<temporary.length)
        {
           tobedecrypted[j][0] = temporary[i];
```

```
            i++;
            tobedecrypted[j][1] = temporary[i];
            i++;
            j++;

        }

        ArrayList<BigInteger> arraydecrypted = gamal.dekrip(p, x, tobedecrypted);
        int[] destringed = gamal.convertbIGIntegers(arraydecrypted);

        String s = "";

        int z;
        for(z=0;z<destringed.length;z++)
        {
           s+=Character.toString ((char)  destringed[z]);
        }
      jTextArea2.setText(s);
```

Cramer-Shoup lebih sulit untuk diimplementasikan sehingga belum bisa diimplementasikan.

V. KESIMPULAN

Secara umum, Cramer Shoup lebih aman daripada ElGamal. Namun, ElGamal lebih mudah untuk diimplementasikan, sehingga lebih banyak yang menggunakan. Untuk memperkuat ElGamal, bisa digunakan fungsi Hash berupa SHA atau MD5 sehingga ElGamal bisa mendekati kekuatan dari Cramer Shoup.

DAFTAR PUSTAKA

[1] Cramer, Shoup. "Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption", 2001 http://www.shoup.net/papers/uhp.pdf

[2] Cramer, Shoup, "A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack", ETH Zurich, 2001 http://knot.kaist.ac.kr/seminar/archive/46/46.pdf

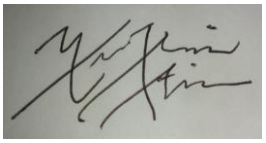
[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2006-2007/Makalah2/Makalah-067.pdf

[4] Munir, RinaldiBahan Kuliah Kriptografi (Kriptografi Kunci Publik dan Algoritma ElGamal), 2013,

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, 20 Mei 2013



Yudhistira
13508105