Implementasi dan Perbandingan Algoritma Kriptografi Kunci Publik

RSA, ElGamal, dan ECC

Vincent Theophilus Ciputra (13513005)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
vincent.theophilusc@gmail.com

***Abstrak*—Saat ini keamanan data merupakan salah satu hal yang sangat diperlukan untuk menjaga privasi. Algoritma-algoritma kriptografi biasa digunakan untuk menjaga keamanan data tersebut. Algoritma kriptografi yang akan dijelaskan pada makalah ini adalah algoritma kriptografi kunci publik, yaitu RSA (Riverst Shamir Adleman), ElGamal, dan ECC (Elliptic Curve Cryptography). Salah satu faktor yang dapat menyatakan baiknya algoritma kriptografi kunci publik ini adalah faktor kecepatan untuk melakukan enkripsi dan dekripsinya. Karena itu, makalah ini berisi tentang perbandingan kecepatan antara algoritma-algoritma tersebut.**

Kata kunci— ECC; ElGamal; Kriptografi; RSA

I. PENDAHULUAN

Algoritma kriptografi digunakan untuk menjaga sistem keamanan data. Ada banyak cara untuk melakukan serangan terhadap algoritma-algoritma kriptografi ini, misalnya dengan menggunakan serangan kriptanalis atau *bruteforce*. Selain mempertimbangkan faktor kekuatan enkripsi dari suatu algoritma kriptografi, faktor kecepatan juga merupakan faktor yang penting untuk memilih algoritma kriptografi yang sebaiknya digunakan. Jika algoritma kriptografi memiliki keamanan yang kuat tetapi proses enkripsinya lambat maka algoritma kriptografi tersebut tidak akan digunakan bila pengguna membutuhkan faktor kecepatan dalam melakukan enkripsi. Contoh yang dapat digunakan di mana kecepatan sangat diperlukan adalah jika algoritma tersebut digunakan pada jaringan komputer yang memiliki arsitektur client-server. Jika jumlah client besar, maka lambatnya kinerja algoritma tersebut akan terlihat dengan jelas karena menambah beban kerja server. Oleh karena itu, faktor kecepatan juga perlu diperhatikan saat memilih algoritma yang akan digunakan.

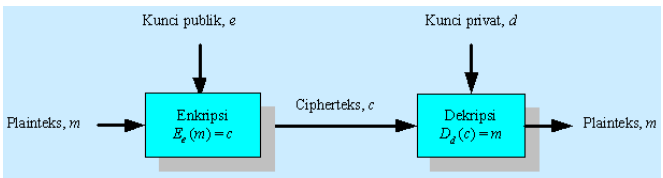
Ada dua teknik dalam melakukan enkripsi dan dekripsi data, yaitu kriptografi simetris dan kriptografi asimetris. Perbedaan dari dua algoritma tersebut terletak pada kuncinya. Algoritma kriptografi simetris menggunakan kunci yang sama dalam melakukan enkripsi dan dekripsinya, sedangkan algoritma kriptografi asimetris menggunakan dua kunci yang berbeda dalam melakukan enkripsi dan dekripsinya, yang biasa disebut kunci publik dan kunci privat.

Pada makalah ini, akan dibahas mengenai algoritma kriptografi kunci publik, atau biasa disebut juga algoritma kriptografi asimetris. Ada tiga algoritma yang terkenal dari algoritma kriptografi kunci publik, yaitu algoritma RSA, ElGamal, dan ECC. Ketiga algoritma diatas akan dibandingkan baik dari segi kecepatan melakukan enkripsi dan dekripsinya, maupun ukuran dari cipherteks yang dihasilkannya.

II. DASAR TEORI

A. *Kriptografi Kunci Publik*

Algoritma kriptografi kunci publik, biasa disebut juga dengan algoritma kriptografi asimetris, dalam melakukan enkripsi dan dekripsinya menggunakan dua kunci, yaitu kunci publik dan kunci privat. Kunci publik adalah kunci yang digunakan saat melakukan enkripsi, sedangkan kunci privat adalah kunci yang digunakan saat melakukan dekripsi. Kunci publik bersifat umum, artinya kunci ini tidak dirahasiakan sehingga dapat dilihat oleh siapa saja. Sedangkan kunci privat adalah kunci yang dirahasiakan dan hanya orang-orang tertentu saja yang boleh mengetahuinya. Keuntungan utama dari algoritma ini adalah memberikan jaminan keamanan kepada siapa saja yang melakukan pertukaran informasi meskipun di antara mereka tidak ada kesepakatan mengenai keamanan pesan terlebih dahulu maupun saling tidak mengenal satu sama lainnya.



Gambar 1. Algoritma kriptografi kunci publik

Algoritma kriptografi kunci publik ini memiliki dua keuntungan, yaitu algoritma ini tidak memerlukan pengiriman kunci privat dan juga jumlah kunci yang digunakan dapat ditekan. Pembangkitan sepasang kunci didasarkan pada persoalan integer sebagai berikut:

1. Pemfaktoran

   Jika diberikan suatu bilangan bulat n, dan dilakukan pemfaktoran n tersebut menjadi faktor primanya.

   Contoh: $10 = 2 * 5$

   $60 = 2 * 2 * 3 * 5$

   $252601 = 41 * 61 * 101$

   $2^{13} - 1 = 3391 * 23279 * 65993 * 1868569 * 1066818132868207$

   Semakin besar nilai n, maka akan semakin sulit untuk memfaktorkannya dan membutuhkan waktu yang sangat lama. Algoritma yang menggunakan prinsip ini adalah algoritma RSA.

2. Logaritma diskrit

   Akan sulit untuk menghitung x sedemikian sehingga $a^x = b \pmod n$.

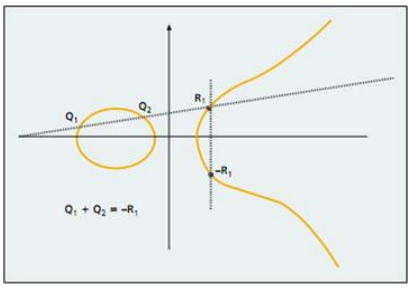
   Contoh: Jika $3^x \equiv 15 \pmod{17}$ maka $x = 6$

   Semakin besar a, b, dan n, maka semakin sulit untuk memfaktorkannya (membutuhkan waktu yang lama).

   Algoritma yang menggunakan prinsip ini adalah algoritma ElGamal dan DSA. Untuk persoalan logaritma diskrit, kebalikan dari persoalan perpangkatan modular. Persamaan $a^x \bmod n$ akan mudah untuk dihitung.

3. Elliptic Curve Discrete Logarithm Problem

   Jika diberikan dua buah titik di suatu kurva eliptik, yaitu P dan Q dan dicari integer n sedemikan sehingga P = n Q.



Gambar 2. ECDLP

   Algoritma yang menggunakan prinsip ini adalah algoritma ECC.

B. *Algoritma RSA (Riverst Shamir Adleman)*

Algoritma RSA ini merupakan algoritma kunci publik yang paling terkenal dan paling banyak aplikasinya. Keamanan dari algoritma RSA ini terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pada algoritma RSA terdapat tiga proses utama, yaitu proses pembentukan kunci, proses enkripsi, dan proses dekripsi.

Prinsip dari proses pembentukan kunci adalah sebagai berikut:

1. Pilih dua bilangan prima, p dan q (rahasia).
2. Hitung $n = pq$.
3. Hitung $\phi(n) = (p-1)(q-1)$.
4. Pilih sebuah bilangan bulat e untuk kunci publik, sebut e relatif prima terhadap $\phi(n)$.
5. Hitung kunci dekripsi, yaitu d, dengan persamaan

   $ed \equiv 1 \pmod{\phi(n)}$ atau $d \equiv e^{-1} \bmod (\phi(n))$.

Hasil dari algoritma di atas adalah kunci publik dengan pasangan (e, n) dan kunci privat dengan pasangan (d, n).

Langkah-langkah dalam proses enkripsi:

1. Nyatakan pesan menjadi blok-blok plainteks : m1, m2, m3 , ... (syarat: $0 < m_i < n - 1$).
2. Hitung blok cipherteks ci untuk blok plainteks pi dengan persamaan $c_i = m_i^e \bmod n$, dalam hal ini, e adalah kunci publik.

Langkah-langkah dalam proses dekripsi:

1. Proses dekripsi dilakukan dengan menggunakan persamaan $m_i = c_i^d \bmod n$, dalam hal ini, d adalah kunci privat.

C. *Algoritma ElGamal*

Keamanan dari algoritma ElGamal terletak pada sulitnya menghitung logaritma diskrit pada bilangan modulo prima yang besar. Pada algoritma ElGamal ini juga terdapat tiga proses, yaitu proses pembentukan kunci, proses enkripsi, dan proses dekripsi.

Algoritma pembangkitan kunci:

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 adalah kunci publik, tripel (y, g, p) dan kunci privat, pasangan (x, p).

Algoritma enkripsi:

1. Susun plainteks menjadi blok-blok m1, m2, ... , (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, karena itu ukuran cipherteks menjadi dua kali ukuran plainteksnya.

Algoritma dekripsi:

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

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

D. Algoritma ECC (Elliptic Curve Cryptography)

Algoritma ECC atau Elliptic Curve Cryptography adalah algoritma kriptografi kunci publik yang lebih baru daripada RSA maupun ElGamal. Algoritma ini memiliki panjang kunci yang lebih pendek dibandingkan panjang kunci algoritma RSA, namun memiliki tingkat keamanan yang sama dengan RSA.

Algoritma ECC adalah salah satu pendekatan algoritma kriptografi kunci publik yang berdasarkan pada struktur aljabar dari kurva ellips pada daerah finite. Elliptic curve juga digunakan pada beberapa algoritma pemfaktoran integer yang juga diaplikasikan dalam kriptografi seperti Lenstra Elliptic Curve Factorization. Beberapa pondasi matematika dari ECC adalah aritmatika modular, groups dan finite field yang di dalamnya terdapat groups, order group an generator, subgroup, finite field, dan *The Discrete Logarithm Problem* (DLP).

Keunggulan dari kriptosistem kurva ellips adalah proses transformasi plainteks menjadi titik-titik dalam kurva ellips sebelum dilakukan enkripsi. Proses enkripsinya dilakukan dengan menggunakan aturan penjumlahan pada kurva ellips. Proses ini tentunya akan memberikan tingkat keamanan yang lebih baik.

Dalam melakukan pembangkitan pasangan kunci publik dan kunci privat, hal yang dilakukan adalah sebagai berikut:

1. Kunci privat = integer x, dipilih dari selang [1, p-1].
2. Kunci publik = titik Q, Q = x . B, dengan B adalah titik basis.

Dalam proses melakukan enkripsi, plainteks M dikode menjadi sebuah titik, P_M dari kurva eliptik. Setelah plainteks menjadi sebuah titik, pengirim plainteks tersebut memilih bilangan acak lain, yaitu k, dari selang [1, p-1]. Cipherteks yang dihasilkan adalah pasangan titik $P_C = [(kB), (P_M + kP_B)]$, dengan P_B adalah kunci publik penerima.

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

$$b \,.\, (kB)$$

Setelah itu, penerima mengurangkan titik kedua dari P_C dengan hasil kali di atas.

$$(P_M + k\,P_B) - [b.(kB)] = P_M + k.(bB) - b.(kB) = P_M$$

Setelah P_M didapatkan, penerima melakukan decode agar P_M kembali menjadi plainteks dan dapat dibaca.

Kriptosistem kurva elips memberikan tingkat keamanan yang lebih baik dibandingkan dengan algoritma asimetris lainnya seperti RSA. Hasil tinjauan pustaka memperlihatkan untuk tingkat keamanan yang sama (MIPS tahun yang sama) kriptosistem kurva elips memerlukan jumlah bit kunci yang jauh lebih sedikit dibandingkan dengan RSA atau DSA. Hal ini tentunya kriptosistem kurva ellips dapat menjadi pilihan yang baik untuk membangun sistem kriptografi yang memiliki tingkat kemanan yang tinggi.

III. PEMBAHASAN

Dari ketiga algoritma RSA, ElGamal, dan ECC, masing-masing algoritma memiliki kelebihan dan kekurangannya. Pada bagian ini, akan dibahas mengenai kelebihan dan kekurangan dari ketiga algoritma tersebut.

Pada algoritma yang pertama, yaitu algoritma RSA, kekuatan algoritmanya ini terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor primanya, dalam hal ini memfaktorkan n menjadi p dan q. Jika n sudah berhasil difaktorkan menjadi p dan q, maka rumus $\phi(n) = (p-1)(q-1)$, sudah bisa dihitung. Selanjutnya, walaupun nilai e diumumkan atau tidak rahasia, perhitungan kunci d tidaklah mudah karena nilai m yang tidak diketahui tetapi dapat dihitung dari persamaan $ed \equiv 1 \pmod{\phi(n)}$.

Kelebihan lain dari algoritma RSA ini terletak pada ketahanannya terhadap berbagai bentuk serangan, terutama serangan brute force. Hal ini dikarenakan kompleksitas dekripsinya yang dapat ditentukan secara dinamis dengan cara menentukan nilai p dan q yang besar pada saat proses pembangkitan pasangan kunci, sehingga dihasilkan sebuah key space yang cukup besar, sehingga tahan terhadap serangan-serangan tersebut.

Algoritma RSA ini juga memiliki beberapa kelemahannya. Ukuran kunci privat yang terlalu besar akan mengakibatkan proses dekripsi yang cukup lambat, apalagi ukuran pesan tersebut semakin besar. Karena ukuran kunci privatnya yang terlalu besar, algoritma RSA lebih lambat daripada algoritma kriptografi simetri, seperti DES dan AES. Oleh karena itu, algoritma RSA lebih baik digunakan untuk mengenkripsi pesan berukuran kecil, yaitu kunci simetri dengan kunci publik menerima pesan dan pesan tersebut dienkripsi dengan menggunakan algoritma simetri yang lebih cepat seperti DES atau AES.

Pada algoritma kedua, yaitu algoritma ElGamal, kekuatan algoritmanya terletak pada kesulitan perhitungan logaritma diskrit pada modulo prima yang besar, sehingga akan sulit untuk menyelesaikan masalah logaritma ini. Algoritma ini memiliki kelebihan dalam melakukan pembangkitan kunci karena menggunakan logaritma diskrit dan metode enkripsi dekripsi dengan proses komputasi yang besar sehingga hasil enkripsinya berukuran dua kali dari ukuran semula.

Selain dari kelebihannya tersebut, algoritma ini memiliki kekurangan karena membutuhkan resource yang besar.

Chiperteks yang dihasilkan dari algoritma ini menjadi dua kali panjang plainteks serta membutuhkan processor yang mampu untuk melakukan komputasi yang besar untuk perhitungan logaritma perpangkatan besar. Dalam melakukan dekripsi, algoritma ini membutuhkan waktu yang lebih lama karena kompleksitas proses dekripsinya yang rumit. Karena ukuran cipheteks yang dihasilkan menjadi dua kali panjang plainteks, maka komputasi yang dibutuhkan pun menjadi dua kali lipat.

Pada algoritma yang ketiga, yaitu algoritma ECC, keuntungan yang didapatkan sama saja dengan algoritma kriptografi lain, seperti dalam hal *confidentiality*, *integrity*, *authentication*, dan *non-repudiation* tetapi algoritma ini menggunakan panjang kunci yang lebih pendek dari algoritma lain. Contohnya dalam melakukan enkripsi kunci algoritma AES sepanjang 128-bit, dengan algoritma ECC hanya menggunakan ukuran kunci 256-bit, tetapi algoritma RSA menggunakan ukuran kunci 3072-bit. Ukuran kunci yang lebih pendek tersebut dapat menghemat *storage* dan *bandwidth* yang digunakan, selain itu dalam proses enkripsi atau dekripsi pun akan menjadi lebih cepat. Tetapi algoritma ini masih memiliki kecepatan yang lebih rendah jika dibandingkan dengan algoritma simetris.

IV. Analisis

Pada penelitian yang dilakukan ini, akan diukur kecepatan enkripsi dan dekripsi dari tiga algoritma kunci publik, yaitu algoritma RSA, ElGamal, dan ECC dan dibuat perbandingannya. Untuk menghitung waktu enkripsi dan dekripsi, akan dibuat juga perangkat lunak yang sudah diimplementasikan ketiga algoritma tersebut dan dihitung waktu enkripsi dan dekripsinya. Implementasi perangkat lunak tersebut dilakukan pada lingkungan dengan spesifikasi sebagai berikut:

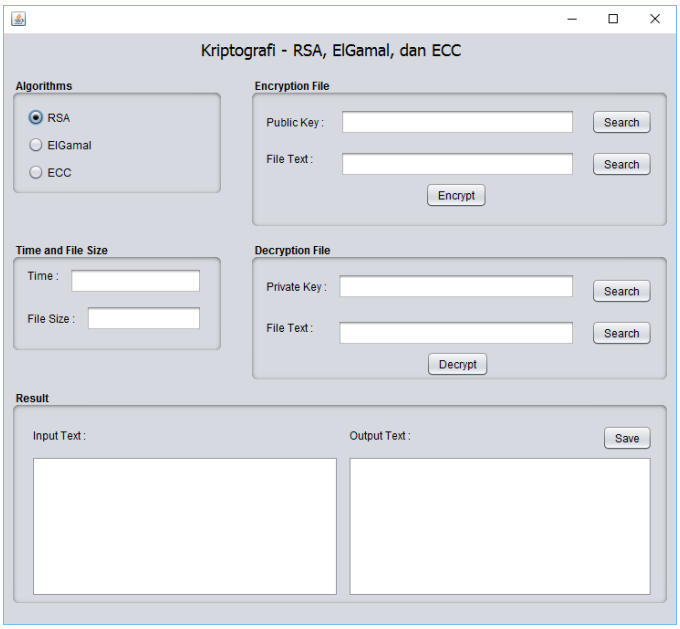
1. Processor: Intel(R) Core(TM) i3-2100 CPU @3.10GHz 3.40GHz
2. RAM: 4.00 GB
3. System Type: 64-bit Operating System
4. Hard Disk: 1TB
5. VGA: NVIDIA GeForce GT 430
6. Operating System: Microsoft Windows 7 Ultimate

Dalam mengukur kecepatan enkripsi dan dekripsi dari ketiga algoritma tersebut, akan dibuat tiga standar dalam setiap eksperimen agar hasil yang didapatkan akan lebih konsisten dan objektif. Standar yang dibuat meliputi parameter sistem, faktor eksperimen, dan prosedur simulasi.

Dalam standar parameter sistem, eksperimen akan dilakukan pada komputer yang sama dengan spesifikasi seperti sudah dijelaskan sebelumnya. Eksperimen ini akan dilakukan beberapa kali dan bisa didapatkan rata-ratanya karena kecepatan eksekusi program dapat berubah-ubah tidak menentu. Hal ini dilakukan agar hasil yang didapatkan lebih konsisten.

Pada standar faktor eksperimen, yang akan dianalisis adalah kecepatan enkripsi dan dekripsi dari setiap algoritma. Selain itu akan dilihat juga ukuran-ukuran cipherteks yang dihasilkan. Ukuran tersebut akan dihitung dalam byte.

Pada standar ketiga yaitu standar prosedur simulasi, simulasi akan dilakukan dengan menggunakan GUI yang sudah dibuat sebelumnya. Program simulasi ini digunakan untuk melakukan enkripsi dan dekripsi dari berbagai ukuran data dan akan dicatat ukuran data tersebut beserta waktu enkripsi atau dekripsinya. Waktu yang dicatat tersebut digunakan untuk mengetahui kecepatan proses enkripsi atau dekripsinya. Pada program yang sudah dibuat ini, pengguna dapat memasukkan pesan yang akan dienkripsi atau dekripsi, dan juga memilih algoritma yang akan digunakan, yaitu algoritma RSA, ElGamal, atau ECC. Hasil keluaran waktu enkripsi atau dekripsi akan ditampilkan melalui GUI dan dapat dibuktikan juga kebenaran algoritma dilihat dari perbandingan hasil dekripsi dengan plainteks aslinya. GUI yang digunakan seperti pada gambar 3 di bawah ini.



Gambar 3. Tampilan GUI program

Dalam melakukan perbandingan ketiga algoritma ini, hasil waktu yang sudah dicatat setiap melakukan enkripsi dan dekripsi, akan dimasukkan ke dalam grafik kartesian yang akan menampilkan waktu enkripsi atau dekripsi dengan ukuran data yang digunakan. Hal tersebut dilakukan agar perbandingan dapat terlihat dengan lebih jelas.

Dari eksperimen yang telah dilakukan, jika ukuran data terlalu kecil, misalnya dibawah 10Kb, waktu dalam melakukan proses enkripsi dan dekripsinya tidak dapat diukur karena ukuran datanya sangat kecil dan waktunya menjadi kurang dari 1ms. Karena itu percobaan yang dilakukan menggunakan ukuran data dari 10Kb sampai 100Kb dengan kelipatan 10. Waktu akan diambil dari rata-rata lima kali percobaan untuk setiap algoritma berbeda dan setiap ukuran data yang berbeda. Hal tersebut dilakukan karena kecepatan eksekusi program bisa berubah-ubah tidak menentu.

Waktu enkripsi dan dekripsi dari algoritma RSA, ElGamal, dan ECC untuk setiap ukuran data beserta ukuran cipherteksnya seperti di dalam tabel 1 dan 2 di bawah ini.

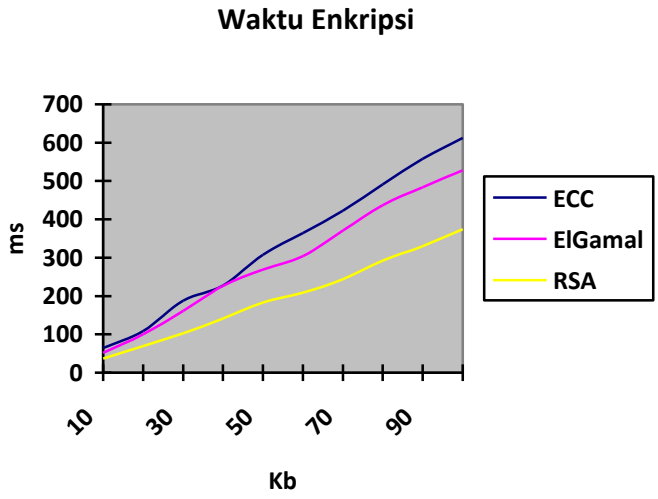
Tabel 1. Tabel waktu enkripsi dan ukuran cipherteks

Ukuran plainteks (Kb)	RSA		ElGamal		ECC	
	Enkripsi (ms)	Ukuran cipherteks (Kb)	Enkripsi (ms)	Ukuran cipherteks (Kb)	Enkripsi (ms)	Ukuran cipherteks (Kb)
10	37	10	52	20	64	20
20	69	20	100	40	109	40
30	103	30	161	61	188	61
40	141	40	226	81	227	81
50	183	50	269	100	307	100
60	209	60	304	121	364	121
70	244	70	371	140	423	140
80	292	80	437	160	491	160
90	330	90	484	180	558	180
100	374	100	528	201	612	201

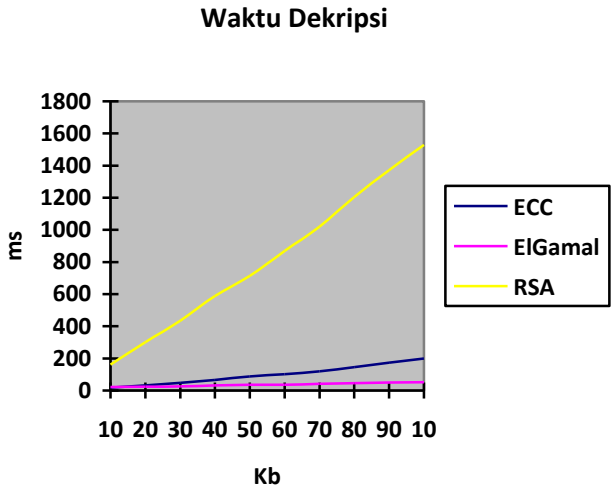
Tabel 2. Tabel waktu dekripsi

Ukuran plainteks (Kb)	RSA	ElGamal	ECC
	Dekripsi (ms)	Dekripsi (ms)	Dekripsi (ms)
10	162	21	17
20	301	24	31
30	434	26	47
40	589	32	65
50	713	35	87
60	869	36	102
70	1021	42	119
80	1204	45	145
90	1371	49	173
100	1528	52	199

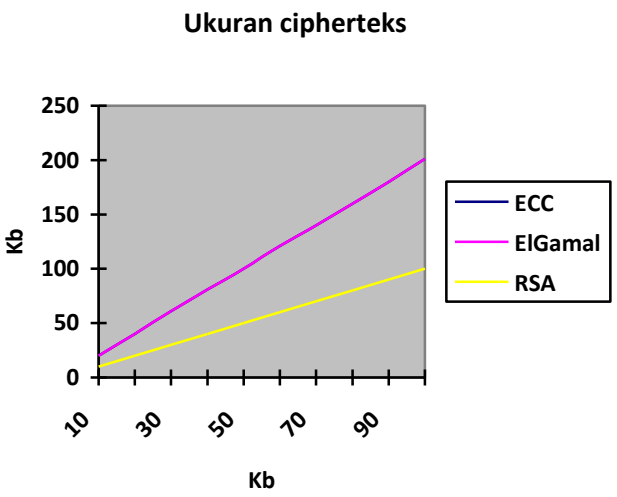
Dari tabel 1 dan tabel 2 yang sudah dibuat diatas, data-data tersebut dimasukkan ke dalam grafik waktu enkripsi dan waktu dekripsi seperti terlihat pada gambar 4 dan gambar 5.



Gambar 4. Grafik waktu Enkripsi dan ukuran data dari ketiga algoritma.



Gambar 5. Grafik waktu Dekripsi dan ukuran data dari ketiga algoritma.



Gambar 6. Grafik ukuran cipherteks

Dari kedua grafik di atas, dapat dilihat bahwa kecepatan proses enkripsi ataupun dekripsi dari ketiga algoritma akan mengalami penurunan secara linier terhadap ukuran data. Hal tersebut dapat terjadi karena semakin besar pesan yang akan dienkripsi ataupun didekripsi, maka semakin lama juga proses enkripsi dan dekripsinya. Selain itu, dengan bertambahnya ukuran pesan yang diberikan, maka perbedaan kecepatan ketiga algoritma tersebut akan semakin terlihat.

Dari grafik pada gambar 4, dapat dilihat bahwa algoritma kriptografi kunci publik yang memiliki kecepatan enkripsi paling cepat adalah algoritma RSA, disusul dengan algoritma ElGamal dan algoritma ECC menempati posisi terakhir atau dapat dibilang yang paling lambat. Dari hasil enkripsi, cipherteks yang dihasilkan oleh algoritma ElGamal dan ECC menjadi dua kali lipat dari plainteks-nya sedangkan algoritma RSA memiliki ukuran cipherteks yang sama dengan plainteks-nya.

Dari grafik dekripsi pada gambar 5, dapat dilihat bahwa algoritma kriptografi yang memiliki kecepatan dekripsi paling cepat adalah algoritma ElGamal, disusul oleh algoritma ECC, dan algoritma RSA menempati posisi terakhir atau yang paling lambat dari ketiga algoritma. Jika seperti dalam proses enkripsi, semakin besar ukuran data maka semakin lama dalam melakukan proses dekripsinya. Tetapi lain halnya dengan ketiga algoritma tersebut. Walaupun hasil cipherteks dari algoritma ElGamal dan ECC menjadi dua kali lipat plainteks-nya, dan dekripsi dilakukan pada cipherteks tersebut, kecepatan dekripsinya tetap lebih cepat daripada algoritma RSA. Hal ini dikarenakan kompleksitas dekripsi algoritma RSA yang dapat ditentukan secara dinamis dengan cara menentukan nilai p dan q yang besar pada saat proses pembangkitan pasangan kunci, sehingga dihasilkan sebuah key space yang cukup besar.

Dari grafik pada gambar 6, dapat dilihat bahwa ukuran cipherteks pada algoritma ElGamal dan ECC menjadi dua kali ukuran plainteks, sedangkan pada algoritma RSA ukuran cipherteksnya sama dengan ukuran plainteks karena itu algoritma RSA tidak terlihat pada grafik dan sejajar pada sumbu x.

V. KESIMPULAN

Algoritma kriptografi dapat digunakan untuk keamanan data dalam pengiriman pesan. Selain mempertimbangkan faktor kekuatan enkripsi dari suatu algoritma kriptografi, faktor kecepatan juga merupakan faktor yang penting untuk memilih algoritma kriptografi yang sebaiknya digunakan. Dari eksperimen yang telah dilakukan mengenai perbandingan tiga algoritma, yaitu algoritma RSA, ElGamal, dan ECC, telah didapatkan pengukuran waktu proses enkripsi dan dekripsi ketiga algoritma tersebut. Dari perbandingan waktu enkripsi ketiga algoritma tersebut, dapat disimpulkan bahwa algoritma RSA menempati posisi pertama sebagai algoritma tercepat dalam melakukan enkripsinya, disusul oleh algoritma ElGamal, dan algoritma ECC menempati posisi terakhir sebagai algoritma terlambat dalam proses enkripsi. Sedangkan urutan kecepatan dekripsi ketiga algoritma tersebut berbeda dari urutan kecepatan enkripsinya, yaitu algoritma ElGamal menempati posisi pertama sebagai algoritma tercepat dalam melakukan proses dekripsinya, disusul oleh algoritma ECC, dan algoritma RSA menempati posisi terakhir sebagai algoritma terlama dalam melakukan proses dekripsi.

Kesimpulan lain yang bisa didapatkan adalah kecepatan enkripsi dan dekripsi dari ketiga algoritma tersebut akan mengalami penurunan seiring dengan bertambahnya ukuran data yang digunakan. Untuk algoritma RSA penurunan kecepatan yang terjadi terlihat sangat signifikan, sedangkan pada dua algoritma lainnya, yaitu ElGamal dan ECC, tidak terlihat secara signifikan.

Acknowledgment

Dengan selesainya penulisan makalah ini, penulis mengucapkan syukur atas rahmat yang telah diberikan oleh Tuhan Yang Maha Esa karena telah selesainya makalah dengan baik dan tepat waktu. Selanjutnya, penulis mengucapkan terima kasih kepada Bapak Rinaldi Munir sebagai dosen mata kuliah IF4020 Kriptografi yang telah memberikan pengetahuan-pengetahuan mengenai algoritma-algoritma kriptografi baik di dalam kelas maupun di luar kelas karena pengetahuan tersebut sangat berguna untuk menyelesaikan makalah ini. Selain itu, penulis juga ingin mengucapkan terima kasih kepada teman-teman yang sudah membantu penulis baik secara langsung maupun tidak langsung sehingga makalah ini dapat selesai dengan baik.

Referensi

[1] Munir, Rinaldi, Kriptografi Kunci Publik, Program Studi Teknik Informatika.

[2] Munir, Rinaldi, Algoritma RSA, Program Studi Teknik Informatika.

[3] Munir, Rinaldi, Algoritma ElGamal, Program Studi Teknik Informatika.

[4] Munir, Rinaldi, ECC, Program Studi Teknik Informatika.

[5] A. Nadeem, "A Performance Comparison of Data Encryption Algorithms, Information and Communication Technologies", in *First Internation Conference on Date of Conference,* 2005.

[6] Simar P. Singh and Raman Maini, "Comparison Data Encryption Algorithm," *International Journal of Computer Science and Communication,* vol 2, no. 1, January-June 2011.

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, 17 Mei 2016

Vincent Theophilus Ciputra
13513005