Perbandingan Algoritma Kriptografi Kunci Publik Schmidt-Samoa dengan Algoritma RSA

Aditya Agung Putra (13510010)[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13510010@std.stei.itb.ac.id

Abstrak - Makalah ini membahas perbandingan dari dua jenis teknik kriptografi kunci publik, yaitu Schmidt-Samoa dan RSA. Perbandingan yang dilakukan berdasarkan pada tahap pembangkitan kunci, analisis keamanan, dan hasil dari implementasi kedua algoritma. Kedua algoritma kriptografi ini dapat dibandingkan secara adil karena keamanan dari kedua algoritma tersebut berdasar pada sulitnya memfaktorkan bilangan yang sangat besar dan merupakan hasil kali dari dua bilangan prima. Perbandingan hasil dari implementasi kedua algoritma dilihat dari performansi keduanya dalam melakukan enkripsi dan dekripsi jenis pesan yang sama.

Kata kunci: algoritma, Schmidt-Samoa, RSA, perbandingan, performansi

I. Pendahuluan

Kriptografi merupakan seni menyembunyikan informasi yang hendak disampaikan ke pihak lain. Kriptografi adalah perpaduan dari matematika dan ilmu komputer yang dikembangkan pada teori informasi dan keamanan informasi. Sesuai dengan definisinya, kriptografi memiliki tujuan untuk menjaga kerahasiaan (*confidentiality*) dan memastikan bahwa pesan yang akan tiba ke penerima adalah asli, tidak direkayasa di perjalanan. Masalah keamanan informasi tersebut dinamakan otentikasi[1].

Seiring perkembangan riset kriptografi, telah dikembangkan teknik kriptografi kunci publik yang menandakan revolusi dari perkembangan teknik-teknik kriptografi. Teknik ini berdasarkan pada penggunaan kunci yang berbeda untuk proses enkripsi dan dekripsi pesan. Proses pemilihan dan penjaminan kerahasiaan kunci berdasarkan pada operasi matematika yang sulit dikerjakan. Hingga kini, telah banyak algoritma kriptografi kunci publik yang digunakan. Beberapa diantaranya adalah RSA, El-Gamal, Diffie-Hellman, Elliptic Curve, Rabin, dan Schmidt-Samoa.

Algoritma Schmidt-Samoa dipublikasikan pada tahun 2006 dan belum banyak dokumentasi dan implementasi pengamanan data yang dilakukan menggunakan algoritma ini. Pada dasarnya algoritma ini memanfaatkan kerumitan dalam menghitung perpangkatan modulo dalam melakukan enkripsi dan dekripsi pada pesan. Kekuatan dari algoritma ini terletak pada sulitnya memfaktorkan suatu bilangan yang sangat besar menjadi suatu bilangan prima dan kuadrat dari bilangan prima lainnya. Dengan demikian, dapat dikatakan bahwa algoritma ini memiliki banyak kesamaan dengan algoritma RSA.

II. Teori Dasar

A. Kriptografi Kunci Publik

Kriptografi kunci publik dapat dianalogikan seperti pesan yang dimasukkan ke dalam kotak surat yang terkunci. Setiap orang dapat memasukkan surat ke dalam kotak tersebut tetapi hanya pemilik kunci kotak surat yang dapat membuka isinya[2]. Sistem kriptografi ini memiliki dua jenis kunci, yaitu kunci publik yang digunakan untuk enkripsi dan kunci privat yang digunakan untuk dekripsi. Karena dua jenis kunci ynag digunakan berbeda maka teknik kriptografi ini disebut juga kriptografi nirsimetri. Misalkan E adalah fungsi enkripsi dan D adalah fungsi dekripsi. Untuk pasangan kunci publik-privat (*e*, *d*) dan pesan-cipherteks (*m*, *c*) berlaku persamaan

$$E_e(m) = c$$
$$D_d(c) = m \tag{2.1}$$

Kriptografi kunci publik didasarkan pada permasalahan-permasalahan matematika yang sulit dipecahkan seperti pemfaktoran dan logaritma diskrit. Contoh masalah pemfaktoran yang sulit adalah mencari faktor prima dari suatu bilangan bulat besar yang merupakan hasil perkalian dari dua bilangan prima besar. Sedangkan contoh masalah logaritma diskrit adalah menentukan nilai *x* yang memenuhi

$$a^x \equiv b \bmod m$$

jika *a*, *b*, dan *m* sudah diketahui.

Teknik kriptografi kunci publik tidak membutuhkan saluran khusus untuk mendistribusikan kunci privat. Kunci publik dapat dikirimkan melalui saluran yang sama dengan pesan. Selain itu jumlah kunci yang dibuat dapat ditekan, yaitu hanya ada dua kunci yang digunakan pada sistem kriptografi ini.

B. Algoritma RSA

Algoritma RSA dibuat oleh tiga orang peneliti dari MIT (Massachussets Institute of Technology) pada tahun 1976 yaitu Ron Rivest, Adi Shamir, dan Leonard Adleman. Keamanan algoritma ini terletak pada sulitnya memfaktorkan suatu bilangan yang merupakan hasil perkalian dari dua bilangan prima. Pemfaktoran ini dilakukan untuk mendapatkan kunci privat. Pembangkitan kunci pada RSA dilakukan berdasarkan algoritma berikut

1. Pilih dua buah bilangan prima berbeda yang sangat besar, p dan q lalu rahasiakan.
2. Hitung $n = p \cdot q$ yang tidak dirahasiakan.
3. Hitung $\phi(n) = (p-1)(q-1)$ yaitu banyaknya bilangan asli di antara 1 dan n yang relatif prima terhadap n. Rumus umum untuk $\phi(n)$ adalah $\phi(n) = n(1-1/p_1)(1-1/p_2)...(1-1/p_n)$ untuk $p_1, p_2, \ldots, p_n$ faktor prima dari n.
4. Pilih kunci publik e yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan kunci privat d yang memenuhi $ed = 1 + k\phi(n)$.

Tahap enkripsi pada algoritma ini dilakukan dengan membagi pesan kedalam beberapa blok pesan sehingga setiap blok pesan memiliki nilai yang lebih kecil dari n. Setelah itu, setiap blok pesan p_i dienkripsi dengan persamaan

$$c_i = p_i^e \bmod n \tag{2.2}$$

Untuk mendekripsi pesan, setiap blok pesan hasil enkripsi c_i didekripsi dengan persamaan

$$p_i = c_i^d \bmod n \tag{2.3}$$

C. Algoritma Schmidt-Samoa

Algoritma Schmidt-Samoa adalah algoritma kriptografi kunci publik yang pertama kali diajukan melalui publikasi ilmiah berjudul "A New Rabin-type Trapdoor Permutation Equivalent to Factoring and Its Application". Keamanan dari algoritma ini terletak pada sulitnya memfaktorkan bilangan bulat besar yang berbentuk p^2q dengan p dan q adalah bilangan prima yang berbeda[3]. Pembangkitan kunci pada algoritma ini dilakukan berdasarkan langkah-langkah berikut

1. Pilih dua bilangan prima besar p dan q lalu hitung $N = p^2q$. N dipilih menjadi kunci publik.
2. Hitung $d = N^{-1} \bmod lcm(p-1, q-1)$.
3. Kunci privat yang digunakan pada sistem adalah (d, p, q).

Tahap enkripsi pada algoritma ini dilakukan dengan membagi pesan kedalam beberapa blok pesan sehingga setiap blok pesan memiliki nilai yang lebih kecil dari pq. Setelah itu, setiap blok pesan m_i dienkripsi dengan persamaan

$$c_i = m_i^N \bmod N \tag{2.4}$$

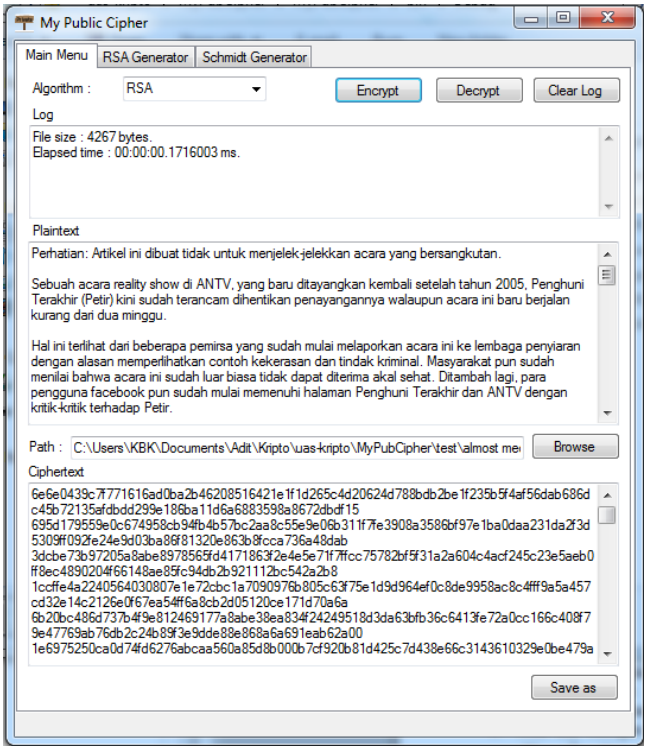
Untuk mendekripsi pesan, setiap blok pesan hasil enkripsi c_i didekripsi dengan persamaan

$$m_i = c_i^d \bmod pq \tag{2.5}$$

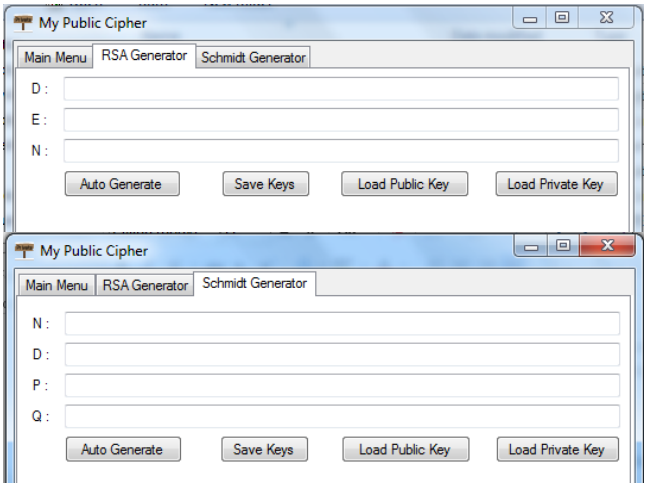
III. Pembahasan

A. Implementasi Kedua Algoritma

Dalam membandingkan kedua algoritma pada makalah ini, dilakukan implementasi terhadap kedua algoritma terlebih dahulu. Hasil implementasi dari kedua algoritma ini adalah suatu program sederhana yang menerima masukan berupa sebuah berkas dan memberikan hasil enkripsi yang direpresentasikan dalam barisan bilangan heksadesimal yang dapat disimpan ke dalam suatu berkas baru. Antarmuka program yang dibuat ditunjukkan pada Gambar III-1 dan III-2.



Gambar III-1 Tampilan program kriptografi kunci publik



Gambar III-2 Antarmuka bagian pembangkit kunci pada program

Implementasi dari program kriptografi yang dibuat dilakukan menggunakan bahasa pemrograman C#, kakas Visual Studio 2012, dan memanfaatkan pustaka BouncyCastle.Crypto.dll untuk menangani operasi pada BigInteger. Tahap enkripsi dan dekripsi dari kedua algoritma dijelaskan sebagai berikut:

1. Enkripsi

   Tahap enkripsi pesan dibagi kedalam tiga tahap. Pertama pesan dibagi kedalam beberapa blok dalam byte sehingga setiap blok sepanjang mungkin namun lebih kecil dari nilai suatu nilai yang ditetapkan. Untuk RSA, nilai yang ditetapkan tersebut adalah *n* yang merupakan perkalian dari dua bilangan prima besar yang dipilih. Untuk Schmidt-Samoa, nilai yang ditetapkan adalah *pq* karena tahap dekripsi blok pada algoritma ini akan menggunakan modulo *pq*.

   Pada tahap kedua, setiap blok dienkripsi menggunakan persamaan yang telah dituliskan pada bagian Teori Dasar untuk masing-masing algoritma. Hasil enkripsi setiap blok dituliskan dalam bilangan heksadesimal. Terakhir, hasil enkripsi semua blok disatukan. Untuk mempermudah skema dekripsi, setiap blok hasil enkripsi dipisahkan dengan pergantian baris pada teks yang dihasilkan.

2. Teknik dekripsi

   Teknik dekripsi pesan juga dibagi kedalam tiga tahap. Tahap pertama adalah pembagian pesan yang akan didekripsi menjadi beberapa upa-teks yang pada teks dipisahkan oleh baris baru. Teks yang akan didekripsi berupa bilangan heksadesimal yang terbagi dalam beberapa baris bergantung pada banyaknya blok pesan yang berhasil dibuat pada saat enkripsi. Tahap berikutnya yaitu mendekripsi setiap upa-teks menggunakan persamaan yang telah dituliskan pada bagian Teori Dasar untuk masing-masing algoritma. Hasil dekripsi setiap upa-teks dituliskan sebagai suatu string. Terakhir, semua hasil dekripsi digabungkan menjadi satu teks utuh.

Skema enkripsi dan dekripsi yang diimplementasikan berhasil bekerja dengan benar untuk masing-masing jenis algoritma. Sebagai contoh, digunakan plainteks berikut ini

```
Pemerintah Kota Surabaya akan meminta pertanggungjawaban penyelenggara acara bagi-bagi es krim. Secara hukum akan dibahas apakah mereka akan dituntut secara pidana atau perdata. Mereka juga akan diminta mengganti kerugiannya, termasuk biaya perawatannya.
```

dan parameter kunci yang dibangkitkan secara acak sebagai berikut

D (RSA)	84111675680965724775741020034004104446263557465867536466745756283874309703493205474478089331923568833138825206021770916978715191429410777126968288130949 09
E (RSA)	89056813336904632375931208591762870029224124336031290680613647952773546197473868880353326332070865764992712729168298162322743705327532612525669395971339 09
N (RSA)	97533348915153697539947699120083347968507997590294479445767023428277075169660156293867778280788035113194336563926457140635993986432904050676830504480275 33
N (Schmidt-Samoa)	10712130797218020703185153613817812856580513311975980221024759554143601219753307828801155072001631266396405660502344042926956026505919528943895426636745563
D (Schmidt-Samoa)	32717306813645035329592949283339008078694950677395056990479706293860704774880210573940079228053070428 9
P (Schmidt-Samoa)	19940915259037382155153968262318779165989172943176 07
Q (Schmidt-Samoa)	26939261810773485278230346742913263637684604524405 87

Tabel III-1 Parameter kunci kriptografi

Didapat hasil enkripsi menggunakan algoritma RSA adalah

```
0d7e14f08807de04369f0bf3fd1898721bb779cd2d2864710823927a3fee341c6868244843ed47c5616ce0a541d6b20323ed8faff1178d759cc50162c4b90ffa

7226514dda5ea7c88a63d52fac9540f7a818360306f3eda820756fa6617e7b5d868318436e65358057c454f6f125645f2d5e4a78f674ad50999b0dbee8626457

00a0876c67834d0fc3ee1dc5d0b1796d51fa1d8660dd6fbc4dd8b8b6f5c9ef3000e458e5004301aab8a3f7a922f0418499535b33167b42edb2a2cbfa977c34ff04
```

49457e2545e42a9ffe287e10c83a3c713765ab7284d2cc899d999ab9a4d15ad2bd2d18798976b96c71c7bf75c245fe9dd079193caeff41b866ff1af7fde4ebed

dan hasil enkripsi menggunakan algoritma Schmidt-Samoa adalah

5a098b7457517df78cdb0adfe74535593c72df767fd8acb2ec86539021b6ead7ef8bdc65b55a51d843e58881c50472f575291b3da0e07f043a696776889959ef

7f0ee60d0525b37298114fa0f125b065e03f33f1eb93b93c85ca909a7f04d1ca43c0d1917dfeb2d9a5c64840e2acffab8b29c8f49cc611c8d5955c1284695ab0

52873e466d5f8990b9518e1b8837898b992a8d53a3228a365b8ed4da7026899a2dd526dbf305d9673ec09c6175ca9b774dd3414b65a635fcca0ddc0316bd16c2

1eae6d6d2c0cbdd8a38050c932cbbd4059a008b5e8af312a619ea2c2417cd06b7e573529ceb307140e4ff75d7a91371abe75e953eda0822ae40dde0953f66b8d

2aeaebae15b1af82cc7392cb2d37f3a03bfc599470bfe682a62f1d0ece1bd814c78f74f7bad55d7472f71253e0b56a8776a089a64a26064bcf96a60d33c3f6e6

14d8b81b6def52b5d9732f87753b0773a98d96c16454817bca00de0554309e03b0a37adfbb4d944f826b884f72f13b45c3312390a5f73d859bbcd503acb7db40

48d169a43acfa8449e63cd70a0651b99e27bd6d3aa6353cf607c650c1dde254bf3a4eeade925e155933391abfe03ab980c2beba50d033dbace7d0b6bd422c8bc

Kedua hasil enkripsi tersebut berhasil didekripsi menjadi plainteks semula.

B. Analisis Kunci Kriptografi

Bilangan N adalah bilangan yang digunakan sebagai modulus pada proses enkripsi dalam kedua algoritma. Untuk ukuran bit N yang sama, pasangan kunci publik pada algoritma RSA, (e, N) lebih panjang dari kunci publik yang digunakan pada algoritma Schmidt-Samoa. Hal tersebut dikarenakan algoritma Schmidt-Samoa hanya menggunakan bilangan N sebagai kunci publik.

Untuk bilangan yang digunakan sebagai kunci privat, algoritma RSA menggunakan pasangan kunci (d, N). Sementara itu, algoritma Schmidt-Samoa menggunakan tripel (d, p, q). d dihitung dari invers modulo N terhadap kelipatan persekutuan terkecil dari p-1 dan q-1 atau LCM(p-1, q-1). Karena nilai tersebut memiliki panjang bit yang lebih pendek dari N yang digunakan, maka panjang keseluruhan dari tripel (d, p, q) lebih pendek dibandingkan pasangan kunci yang digunakan dalam algoritma RSA. Dengan demikian, dapat dikatakan algoritma Schmidt-Samoa memerlukan kapasitas yang lebih kecil untuk menyimpan kunci kriptografi yang digunakan untuk tingkat keamanan yang setara dengan RSA.

Dalam hal keamanan kunci, kedua algoritma sama-sama aman selama faktor dari N tidak diketahui. Walaupun nilai N yang digunakan pada algoritma Schmidt-Samoa merupakan perkalian dari kuadrat bilangan prima dan bilangan prima lain, p^2q tetap tidak lebih kuat dari RSA. Hal ini dikarenakan N yang digunakan pada kedua algoritma hanya memiliki dua buah faktor prima yang berbeda.

C. Analisis Performansi Algoritma

Pengujian performasi kedua algoritma dilihat dari waktu enkripsi dan dekripsi juga panjang cipherteks yang dihasilkan oleh kedua algoritma. Pengujian dilakukan menggunakan panjang bit kunci publik yang sama. Teks yang coba dienkripsi dan dekripsi oleh masing-masing algoritma berupa sepuluh teks dengan panjang yang bervariasi. Pengujian performansi dilakukan pada lingkungan dengan spesifikasi sebagai berikut:

Sistem Operasi : Windows 7 32 bit

Prosesor : Intel(R) Core(TM) i7-4770

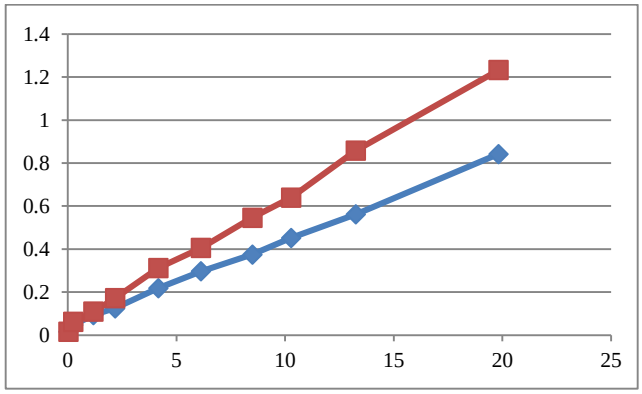
RAM : 8 GB

Sedangkan daftar berkas teks yang digunakan dalam pengujian adalah sebagai berikut

Nama	Ukuran dalam byte
shortest.txt	24
shorter.txt	254
short.txt	1203
not too short.txt	2220
almost medium.txt	4267
medium.txt	6273
long.txt	8692
longer.txt	10527
and longer.txt	13560
longest.txt	20299

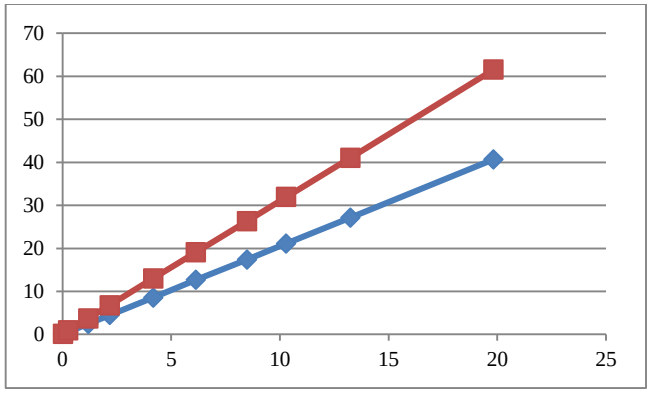
Tabel III-2 Ukuran berkas uji

Perbandingan waktu enkripsi yang dilakukan oleh kedua algoritma digambarkan dalam grafik pada Gambar III-3. Waktu yang dibutuhkan algoritma Schmidt-Samoa ditunjukkan dengan garis berwarna merah sedangkan algoritma RSA oleh garis biru. Sumbu x pada grafik menandakan ukuran teks dalam KB dan sumbu y menyatakan waktu yang dibutuhkan dalam satuan detik. Grafik menunjukkan bahwa untuk ukuran data yang sama, algoritma Schmidt-Samoa memerlukan waktu yang relatif lebih lama. Namun perbedaan waktu yang dibutuhkan tidaklah signifikan karena panjang bilangan yang digunakan dalam perpangkatan modulo pada kedua algoritma adalah sama. Apa yang membuat algoritma RSA lebih cepat pada percobaan ini adalah karena algoritma RSA membagi teks kedalam jumlah blok yang lebih sedikit dibandingkan skema enkripsi pada algoritma Schmidt-Samoa. Hal ini akan lebih dijelaskan pada perbandingan berikutnya.



Gambar III-3 Perbandingan kecepatan enkripsi dari kedua algoritma

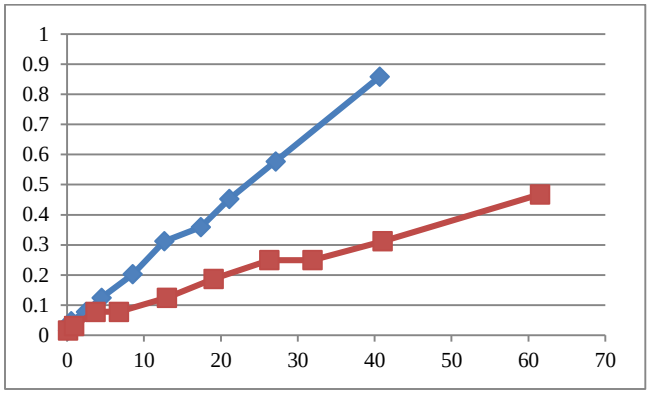
Untuk hasil enkripsi kedua algoritma, perbandingan besar berkas yang dihasilkan digambarkan dalam grafik pada gambar III-4. Besar berkas yang terenkripsi oleh algoritma RSA digambarkan dalam garis biru sementara algoritma Schmidt-Samoa dalam garis merah. Ukuran berkas yang dihasilkan ditunjukkan dalam ukuran KB.



Gambar III-4 Perbandingan ukuran hasil enkripsi kedua algoritma

Terlihat bahwa hasil enkripsi dengan algoritma Schmidt-Samoa berupa teks berukuran lebih besar. Hal ini dikarenakan blok cipherteks yang dihasilkan oleh algoritma ini lebih banyak. Banyak blok cipherteks yang dihasilkan disebabkan oleh ukuran pembagian blok yang lebih kecil pada algoritma Schmidt-Samoa. Pada ukuran teks yang sama, dalam enkripsi pada algoritma RSA setiap blok yang terbagi memiliki ukuran kurang dari N. Sedangkan pada algoritma Schmidt-Samoa setiap blok berukuran kurang dari pq. Karena $N > pq$ maka blok yang dihasilkan oleh algoritma Schmidt-Samoa lebih banyak dibandingkan algoritma RSA.

Untuk proses dekripsi dari kedua algoritma, perbandingan waktu yang digunakan digambarkan dalam grafik pada gambar III-5. Waktu yang dibutuhkan algoritma Schmidt-Samoa ditunjukkan dengan garis berwarna merah sedangkan algoritma RSA oleh garis biru. Sumbu x pada grafik menandakan ukuran teks dalam KB dan sumbu y menyatakan waktu yang dibutuhkan dalam satuan detik.



Gambar III-5 Perbandingan kecepatan dekripsi dari kedua algoritma

Terlihat jelas bahwa waktu yang dibutuhkan untuk melakukan dekripsi pesan pada algoritma Schmidt-Samoa lebih singkat dibandingkan algoritma RSA. Hal ini dikarenakan operasi perpangkatan modulo pada algoritma Schmidt-Samoa melibatkan modulus yang lebih kecil. Dengan demikian, untuk tingkat keamanan yang sama algoritma Schmidt-Samoa memberikan hasil dekripsi secara lebih cepat dibandingkan algoritma RSA.

Perbandingan kedua algoritma ini pada penelitian berikutnya masih dapat dikembangkan. Salah satu penelitian lebih lanjut dapat membahas performansi kedua algoritma saat mengenkripsi atau mendekripsi berkas yang tidak berbentuk teks seperti berkas PDF ataupun aplikasi dengan ekstensi .exe.

IV. Kesimpulan

Dari uraian sebelumnya mengenai perbandingan algoritma RSA dan Schmidt-Samoa, dapat ditarik kesimpulan-kesimpulan berikut ini:

1. Pada program yang digunakan, kedua algoritma berhasil diimplementasikan secara baik dan benar.
2. Keamanan kedua algoritma sama-sama terletak pada sulitnya memfaktorkan bilangan besar yang didapat dari perkalian bilangan-bilangan prima besar.
3. Untuk tingkat keamanan yang sama, algoritma Schmidt-Samoa membutuhkan pasangan kunci dengan ukuran yang lebih kecil dibandingkan pasangan kunci pada algoritma RSA.
4. Kecepatan kedua algoritma dalam melakukan enkripsi teks sebanding karena operasi yang dilakukan pada kedua algoritma adalah perpangkatan modulo dengan bilangan yang sama panjangnya.
5. Algoritma Schmidt-Samoa membagi plainteks kedalam blok-blok yang lebih banyak sehingga membutuhkan waktu enkripsi yang lebih lama. Hal ini juga membuat algoritma ini mnghasilkan cipherteks yang lebih pendek dibandingkan hasil enkripsi dari algoritma RSA.
6. Untuk tingkatan keamanan yang sama, proses dekripsi pada algoritma Schmidt-Samoa lebih cepat.

V. Referensi

[1] Stallings, William. *Cryptography and Network Security Principles and Practice 5th Edition*. New Jersey: Pearson Prentice Hall, 2011.

[2] Munir, Rinaldi. *Kriptografi*. Bandung: Penerbit Informatika, 2006.

[3] K. Schmidt-Samoa, "A New Rabin-type Trapdoor Permutation Equivalent to Factoring and Its Applications," *Electronic Notes in Theoretical Computer Science*, vol. 157, no. 3, pp. 79-94, 2006.

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, 15 Mei 2014

Aditya Agung Putra/13510010