Perbandingan Algoritma RSA dan Rabin

Tadya Rahanady H - 13509070
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
tadtedtod@gmail.com

***Abstrak*—RSA adalah salah satu metode enkripsi dari *asymmetric cipher*. Penggunaan metode *assymetric cipher* ini terbukti lebih sulit dipecahkan daripada penggunaan *symmetric cipher*. Algoritma ini berdasarkan sulitnya pemfaktoran dari bilangan integer yang sangat besar. Pengguna dari RSA membuat dan mempublikasikan hasil dari dua buah bilangan prima yang sangat besar. Setiap orang dapat menggunakan kunci publik untuk mengenkripsi sebuah pesan. Jika kunci tersebut cukup besar, hanya orang dengan pengetahuan mengenai faktor primanya yang dapat memecahkan kode tersebut.**

Rabin adalah metode enkripsi dari *asymmetric cipher* yang mirip seperti RSA, yaitu terkait dengan sulitnya pemfaktoran bilangan. Algoritma Rabin merupakan algoritma varian RSA. Fungsi dasar algoritma Rabin mirip dengan fungsi dasar dari algoritma RSA. Hanya saja komputasi algoritma Rabin lebih sederhana dibandingkan algoritma RSA. Dalam proses enkripsi dan dekripsi, algoritma ini terbilang lebih baik jika dibandingkan dengaan RSA.

Kata Kunci— kriptografi, asymmetric key, public key, private key, cipher, enkripsi, dekripsi, assymetric key, public-key cryptography, RSA, dan Rabin

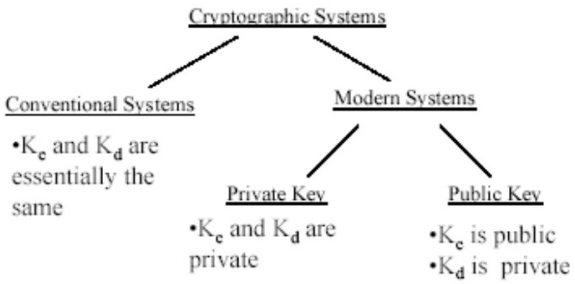
I. Pendahuluan

Kata *cryptography* berasal dari bahasa Yunani yang artinya *secret writing.* Kriptografi adalah suatu praktek dan ilmu dari teknik untuk komunikasi yang aman dimana adanya kehadiran dari pihak ketiga. Secara lebih umum, ilmu ini berisi tentang bagaimana membangun dan menganalisis protokol yang menanggulangi keterlibatan dari orang ketiga yang berhubungan dengan berbagai aspek dalam keamanan informasi, seperti kerahasiaan, keutuhan data, dan keotentikan. Kriptografi modern merupakan gabungan dari disiplin ilmu matematika, sains komputer, dan elektroteknik. Definisi baru yang diajukan oleh Schneier yaitu "Kriptografi adalah seni dan ilmu untuk menjaga agar suatu pesan tetap aman".

Bentuk paling awal dari kriptografi memerlukan tidak lebih dari alat untuk menulis dan juga kertas, karena kebanyakan orang masih belum dapat membaca. Cipher klasik yang paling umum digunakan adalah *transposition ciphers,* yang mengubah urutan huruf-huruf yang ada di dalam pesan, dan *substitution ciphers,* yang secara sistematis mengganti huruf ataupun kumpulan huruf dengan huruf atau kumpulan huruf lainnya. *Substitution cipher* paling awal adalah *Caesar cipher*, dimana setiap huruf yang ada digeser dan menggantikan huruf lainnya.

Dengan perkembangan komputer digital dan elektronik membantu dalam kriptanalisis, memungkinkan untuk dibuatnya *cipher* yang lebih kompleks. Terlebih lagi, komputer memungkinkan enkripsi data jenis apapun yang direpresentasikan dalam format biner. Penggunaan komputer telah menggantikan kriptografi linguistik, baik untuk pembuatan desain *cipher* dan kriptanalisis. Banyak *cipher* yang dapat dikarakteristikkan berdasarkan operasinya dalam urutan bit biner (terkadang dalam grup atau blok), yang secara umum memanipulasi karakter huruf dan digit secara langsung. Selain daripada itu, komputer juga membantu dalam proses kriptanalisis.

Meskipun demikian, perkembangan *cipher* modern yang baik jauh meninggalkan perkembangan kriptanalisis, biasanya terjadi dikarenakan penggunaan *cipher* yang berkualitas sangat efisien, sementara memecahkannya memerlukan usaha yang besar, dan biasanya jauh lebih besar dari usaha yang dibutuhkan untuk pemecahan *cipher* klasik, membuat kriptanalisis sangat tidak efisien dan tidak dapat dipraktekkan secara efektif. Beberapa contoh dari *cipher* modern adalah RSA, ElGamal, dan Rabin.



Gambar 1. Kriptografi

II. Public-Key Cryptography

Kriptografi kunci publik mengacu pada sistem kriptogafi yang memerlukan dua kunci yang berbeda, satu untuk mengunci atau mengenkripsi *plaintext*, dan satu lagi untuk membuka atau mendekripsi *cyphertext.* Setiap kunci hanya bisa melakukan salah satu fungsi saja. Salah satu kunci akan disebarkan, yang disebut *public key*, sedangkan kunci yang satunya akan dirahasiakan atau disebut *private key.*

Kriptografi kunci publik menggunakan algoritma kunci asimetrik, yang lebih dikenal sebagai *asymmetric key cryptography.* Algoritma tersebut memiliki properti *public key* dan *private key* dimana salah satu kunci tidak memiliki informasi dari kunci lainnya. *Public key* digunakan untuk mengubah suatu pesan menjadi bentuk yang tidak dapat dibaca dan dapat didekripsi menggunakan *private key* yang cocok.

Penggunaan sistem ini harus membuat *public key* dan *private key* yang saling berpasangan secara matematis. Dengan menyebarkan *public key,* pembuat kunci memberikan hak pada siapapun yang mendapatkan *public key* untuk mengirim pesan aman yang hanya bisa dibaca oleh si pembuat kunci.

Asymmetric key algorithms berbeda dengan *symmetric key algorithms* dalam proses kerjanya, dimana pengirim pesan lebih mudah untuk mengenkripsi menggunakan *public key* dan penerima pesan untuk mendekripsi menggunakan *private key,* tetapi sangat sulit bagi orang lain untuk menebak *private key* berdasarkan pengetahuan mereka mengenai *public key.*

Teknik yang digunakan dalam kriptografi kunci publik adalah penggunaan algoritma *asymmetric key* dimana kunci yang digunakan untuk enkripsi adalah kunci yang berbeda dengan kunci yang digunakan untuk dekripsi. Tiap pengguna memiliki sepasang kunci kriptografik, *public key* untuk enkripsi dan *private key* untuk dekripsi. *Public key* untuk mengenkripsi disebarkan secara luas, sedangkan *private key* hanya akan diketahui oleh penerima pesan. Pesan akan dienkripsi oleh *public key* pengirim dan hanya akan bisa didekripsi oleh *private key* yang benar. Kedua kunci berhubungan secara matematis, dan algoritma yang dapat menghasilkan pasangan kedua kunci tersebut mulai berkembang pada pertengahan tahun 1970.

Dua permasalahan matematika yang sering dijadikan dasar pencarian sepasang kunci pada kriptografi kunci publik, yaitu:

1. Pemfaktoran
   Diberikan bilangan bulat n. Faktorkan menjadi faktor primanya
   Contoh:
   10 = 2x5
   60 = 2 x 2 x 3 x 5
   252601 = 41 x 61 x 101
   $2^{13} - 1$ = 3391 x 23279 x 65993 x 1868569 x 1066818132868207
   Semakin besar n, semakin sulit memfaktorkan (butuh waktu yang sangat lama). Algoritma yang menggunakan prinsip ini: RSA.
2. Logaritma diskrit
   Temukan x sedemikian sehingga $ax \equiv b \pmod n$ sulit dihitung.
   Contoh: jika $3x \equiv 15 \pmod{17}$ maka x = 6.
   Semakin besar a, b, n semakin sulit memfaktorkan (butuh waktu yang lama). Algoritma yang menggunakan prinsip ini: ElGamal dan DSA.

Jika dibandingkan dengan *symmetric key algorithms*, *asymmetric key algorithms* memiliki kelebihan sebagai berikut:

1. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomuniaksi. Tidak ada kebutuhan mengirim kunci privat sebagaimana pada sistem simetri.
2. Pasangan kunci publik/kunci privat tidak perlu diubah, bahkan dalam periode waktu yang panjang.
3. Dapat digunakan untuk mengamankan pengiriman kunci simetri.
4. Beberapa algoritma kunci-publik dapat digunakan untuk memberi tanda tangan digital pada pesan.

Meskipun penggunaan *asymmetric key algorithms* dapat dikatakan cukup baik, tetapi metode ini memiliki cukup banyak kelemahan, antara lain adalah:

1. Enkripsi dan dekripsi memakan waktu yang lama, karena dekripsi dan enkripsi melibatkan bilangan-bilangan yang besar dan operasi-operasi perpangkatan yang besar.
2. Ukuran *ciphertext* jauh lebih besar dibandingkan *plaintext.*
3. Ukuran kunci relatif besar.
4. *Public key* bisa digunakan siapa saja dan tidak memberikan otentikasi pengirim.

III. RSA

Salah satu algoritma kunci publik yang paling terkenal dan memiliki paling banyak aplikasi adalah algoritma RSA. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT pada tahun 1976, yaitu: Ron Rivest, Adi Shamir, dan Leonard Adleman. Kelebihan dari keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin.

Algoritma RSA memiliki besaran-besaran sebagai berikut:

2. p dan q, bilangan prima (rahasia)
3. n = p . q (tidak rahasia)

4. $\Phi(n) = (p-1)(q-1)$ (rahasia)
5. e (kunci enkripsi) (tidak rahasia)
6. d (kunci dekripsi) (rahasia)
7. m (plainteks) (rahasia)
8. c (chiperteks) (tidak rahasia)

Algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa

$$a^{\Phi(n)} \equiv 1 \pmod{n} \quad (1)$$

dengan syarat:

1. a harus relatif prima terhadap n
2. $\Phi(n)$ = fungsi yang menentukan berapa banyak dari bilangan-bilangan 1, 2, 3, …, n yang relatif prima terhadap n.

Berdasarkan sifat $ak \equiv bk \pmod{n}$ untuk k nilangan bulat ≥ 1, maka persamaan (1) di atas dapat ditulis menjadi

$$a^{k\Phi(n)} \equiv 1^k \pmod{n} \quad (2)$$

atau

$$a^{k\Phi(n)} \equiv 1 \pmod{n} \quad (3)$$

Bila a diganti dengan m, maka persamaan (3) dapat ditulis menjadi

$$m^{k\Phi(n)} \equiv 1 \pmod{n} \quad (4)$$

Berdasarkan sifat $ac \equiv bc \pmod{n}$ maka bila persamaan (4) dikalikan dengan m menjadi

$$m^{k\Phi(n)+1} \equiv m \pmod{n} \quad (5)$$

yang dalam hal ini relatif prima terhadap n. Misalkan e dan d dipilih sedemikian sehingga

$$e \cdot d \equiv 1 \pmod{\Phi(n)} \quad (6)$$

atau

$$e \cdot d \equiv k\Phi(n) + 1 \quad (7)$$

Sulihkan persamaan (7) ke dalam persamaan (5) menjadi

$$m^{e \cdot d} \equiv m \pmod{n} \quad (8)$$

Persamaan (8) dapat ditulis kembali menjadi

$$(m^e)^d \equiv m \pmod{n} \quad (9)$$

yang artinya, perpangkatan m dengan e diikuti dengan perpangkatan dengan d menghasilkan kembali m semula. Berdasarkan persamaan (9), maka enkripsi dan dekripsi dirumuskan sebagai berikut:

$$E_e(m) \equiv c \equiv m^e \bmod n \quad (10)$$

$$D_d(c) \equiv m \equiv c^d \bmod n \quad (11)$$

Karena e . d = d . e, maka enkripsi diikuti dengan dekripsi ekivalen dengan dekripsi diikuti enkripsi:

$$D_d(E_e(m)) = E_e(D_d(m)) = m^d \bmod n \quad (12)$$

Oleh karena $md \bmod n \equiv (m + jn)d \bmod n$ untuk sembarang bilangan bulat j, maka tiap plainteks m, m +n, m + 2n, …, menghasilkan cipher yang sama. Dengan kata lain, transformasinya dari banyak ke satu. Agar transformasinya satu ke satu, maka m harus dibatasi dalam himpunan {0, 1, 2, …, n-1} sehingga enkripsi dan dekripsi tetap benar seperti dalam persamaan (10) dan (11).

Algoritma untuk mencari pasangan kunci:

1. Pilih dua buah bilangan prima sembarang, p dan q.
2. Hitung n = p . q (sebaiknya $p \neq q$, sebab jika p = q maka n = p2 sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n).
3. Hitung $\Phi(n) = (p-1)(q-1)$.
4. Pilih *public key*, e, yang relatif prima terhadap $\Phi(n)$.
5. Temukan *private key* dengan menggunakan persamaan (6), yaitu $e \cdot d \equiv 1 \pmod{\Phi(n)}$. Perhatikan bahwa $e \cdot d \equiv 1 \pmod{\Phi(n)}$ ekivalen dengan $e \cdot d = 1 + k\Phi(n)$, sehingga secara sederhana d dapat dihitung dengan

$$d = 1 + k\Phi(n) / e \quad (13)$$

Hasil dari algoritma di atas adalah:

1. *Public key* adalah pasangan (e,n)
2. *Private key* adalah pasangan (d,n)

n tidak bersifat rahasia, sebab ia diperlukan pada perhitungan enkripsi/dekripsi.

Kekuatan algoritma *RSA* terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor-faktor prima, yang dalam hal ini $n = a \times b$.

Sekali *n* berhasil difaktorkan menjadi *a* dan *b*, maka $\Phi(n) = (a - 1) \times (b - 1)$ dapat dihitung. Selanjutnya, karena kunci enkripsi *e* diumumkan (tidak rahasia), maka kunci dekripsi *d* dapat dihitung dari persamaan $ed \equiv 1 \pmod{n}$.

Penemu algoritma *RSA* menyarankan nilai *a* dan *b* panjangnya lebih dari 100 digit. Dengan demikian hasil kali $n = a \times b$ akan berukuran lebih dari 200 digit. Jika berusaha untuk memecahkan kode, usaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun.

Panjang desimal *n*	Panjang *n* dalam bit (perkiraan)	Perolehan Data	MIPS-Year
100	332	April 1991	7
110	365	April 1992	75
120	398	June 1993	830
129	428	April 1994	5000
130	431	April 1996	500

Ket: *MIPS-Year = million instructions-per-second processor running for one year*, setara dengan eksekusi 3 x 10^{13} instruksi
Prosesor Pentium 200 MHz setara dengan mesin 50-MIPS

Tabel 1. MIPS-Year

Kelemahan dari RSA dijabarkan sebagai berikut:

1. *RSA* lebih lambat daripada algoritma kriptografi kunci-simetri seperti *DES* dan *AES*.
2. Dalam praktek, *RSA* tidak digunakan untuk mengenkripsi pesan, tetapi mengenkripsi kunci simetri (kunci sesi) dengan kunci publik penerima pesan.
3. Pesan dienkripsi dengan algoritma simetri seperti DES atau AES.
4. Pesan dan kunci rahasia dikirim bersamaan.
5. Penerima mendekripsi kunci simetri dengan kunci privatnya, lalu mendekripsi pesan dengan kunci simetri tersebut.

IV. RABIN

Algoritma Rabin pertama kali diperkenalkan pada tahun 1979 oleh Michael O. Rabin. Algoritma Rabin merupakan metode kriptografi asimetris pertama dimana untuk mendapatkan kembali plainteks dari cipherteks yang ada terbukti sama sulitnya seperti proses pemfaktoran.

a. Pembangkitan kunci

Sama seperti sistem kriptografi asimetri lainnya, Rabin juga menggunakan sistem kunci publik dan kunci privat. Kunci publik digunakan pada proses enkripsi dan dapat diketahui oleh semua pihak (tidak rahasia), sementara kunci privat digunakan oleh penerima pesan untuk dekripsi dan bersifat rahasia.

Algoritma pembangkitan kuncinya adalah sebagai berikut:

1. Pilih dua buah bilangan prima besar sebarang yang saling berbeda (p dan q).
2. Hitung n = p.q. n adalah kunci publik. Bilangan prima p dan q adalah kunci privat.

Untuk mengenkripsi pesan hanya dibutuhkan kunci publik n, sedangkan untuk dekripsi, dibutuhkan bilangan p dan q sebagai kunci privat.

b. Metode enkripsi

Algoritma Rabin merupakan algoritma kriptografi kunci publik, maka enkripsi dilakukan hanya dengan menggunakan kunci publik yang dapat diketahui oleh semua orang, namum proses dekripsi hanya dapat dilakukan dengan menggunakan kunci privat oleh orang yang bersangkutan. Proses enkripsi pada teknik Rabin sangat sederhana. Proses enkripsi tersebut dapat dituliskan dengan rumus berikut:

$C = P^2 \bmod n$

Keterangan:
C : Cipherteks
P : Plainteks
n : Kunci publik

Karena proses enkripsi yang sederhana dari Rabin ini, menyebabkan waktu yang digunakan relatif singkat karena tidak memiliki proses yang rumit. Kesederhanaan ini merupakan salah satu keuntungan yang dimiliki oleh teknik Rabin untuk menghadapi keterbatasan sumber daya yang ada pada media kriptografi.

c. Metode dekripsi

Proses dekripsi pada teknik Rabin dilakukan dengan menggunakan sebuah rumus sederhana, akan tetapi dengan tambahan teorema Chinese remainder. Teorema ini digunakan untuk mendapatkan plainteks yang benar. Namun poin penting dari metode ini adalah bahwa metode Rabin tidak menghasilkan jawaban plainteks tunggal. Jawaban yang dihasilkan dari dekripsi metode Rabin ini terdiri dari 4 kemungkinan jawaban, dan tidak menghasilkan satu jawaban yang pasti. Berikut adalah contoh algoritma proses dekripsi teknik kriptografi Rabin:

```
Dekripsi(p,q,C)
{
a1 ← +(C(p+1)/4) mod p
a2 ← -(C(p+1)/4) mod p
b1 ← +(C(q+1)/4) mod q
b2 ← -(C(q+1)/4) mod q

// Chinese_Rem adalah fungsi yang memanggil fungsi untuk Chinese Remainder

P1 ← Chinese_Rem(a1,b1,p,q)
P2 ← Chinese_Rem(a1,b2,p,q)
P3 ← Chinese_Rem(a2,b1,p,q)
P4 ← Chinese_Rem(a2,b2,p,q)

return P1,P2,P3,P4
}
```

Metode Rabin akan selalu menghasilkan empat kemungkinan hasil, yang diberikan sebagai hasil dari dekripsi terhadap pesan rahasia. Kemudian penerima pesan harus dapat menentukan pesan mana yang sebenarnya dari keempat hasil dekripsi tersebut. Meskipun menghasilkan empat pesan berbeda pada akhirnya, namun

penerima pesan relatif dapat memilih pesan mana yang benar dengan mudah karena pesan yang benar seharusnya dapat terlihat jelas jika dibandingkan dengan ketiga hasil dekripsi yang lain. Akan tetapi, jika pesan yang ditulis merupakan urutan dari nilai-nilai numerik, hal ini akan menjadi permasalahan yang harus diselesaikan menggunakan sebuah skema untuk menghilangkan ambiguitas.

V. Perbandingan Keamanan Kedua Algoritma

Untuk membandingkan tingkat keamanan dari kedua algoritma tersebut dapat dilihat kelebiha beserta kekurangan dari masing-masing algoritma tersebut :

A. Kelebihan dan Kekurangan RSA

Kekuatan dari algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor-faktor primanya, yang dalam hal ini n = a x b. Sekalipun n berhasil difaktorkan menjadi a dan b, maka (n) = (a – 1)x(b – 1) dapat dihitung. Selanjutnya, karena kunci enkripsi e diumumkan (tidak rahasia), maka kunci dekripsi d dapat dihitung dari persamaan ed 1 (mod n). Penemu dari algoritma ini menyarankan agar nilai a dan b panjangnya lebih dari 100 digit. Dengan demikian hasil kali n = a x b akan berukuran lebih dari 200 digit. Dengan demikian hasil perkalian n = a x b akan menghasilkan angka yang berukuran lebih dari 200 digit. Jika dihitung, maka usaha untuk mencari faktor bilangan berukuran 200 digit akan membutuhkan waktu komputasi selama 4 miliar tahun dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik.

Akan tetapi disamping kelebihannya itu, RSA memiliki juga beberapa kekurangan. Algoritma RSA lebih lambat daripada algoritma Rabin ataupun algoritma kriptografi kunci simetri seperti DES dan AES. Dalam prakteknya juga, RSA tidak digunakan untuk mengenkripsi pesan, tetapi mengenkripsi kunci simetri dengan kunci publik penerima pesan.

B. Kelebihan dan Kekurangan Rabin

Keuntungan besar dari metode Rabin adalah bahwa plainteks acak dapat didapatkan kembali sepenuhnya dari cipherteks hanya jika pemecah kode mampu secara efisien memfaktorkan kunci publik n.

Telah terbukti bahwa memecahkan algoritma Rabin sama seperti pemecahan masalah pemfaktoran integer, yang sedikit berbeda dibandingkan RSA. Dengan demikian, sistem Rabin lebih aman dalam hal ini daripada RSA, dan akan tetap demikian sampai solusi umum untuk masalah faktorisasi ditemukan.

Dikarenakan solusi untuk permasalahan dari pemfaktoran sedang dicari dalam berbagai bidang yang berbeda, solusi apa pun dengan cepat akan tersedia bagi komunitas ilmiah secara keseluruhan. Akan tetapi, sebuah solusi telah lama ditemukan, dan masalah pemfaktoran secara praktik tidak memiliki solusi. Dengan demikian, penyerang tidak akan memiliki kesempatan untuk memecahkan kode. Metode ini telah terbukti aman terhadap *chosen plaintext attacks*.

Namun, penyerang aktif dapat memecahkan sistem menggunakan *chosen ciphertext attack* , yang telah terbukti secara matematis.

C. Perbandingan Parameter Kedua Algoritma

Karena algoritma Rabin merupakan varian dari RSA, dia memiliki fungsi dasar yang cukup mirip. Pada RSA, proses pembangkitan kunci sedikit lebih rumit jika dibandingkan dengan Rabin karena menggunakan nilai yang lebih banyak. Pembangkitan kunci pada RSA menggunakan teknik pemfaktoran yang membutuhkan 3 nilai, yaitu p, q, dan N. Sedangkan proses pembangkitan kunci pada algoritma Rabin hanya menggunakan 2 nilai saja, yaitu p dan q yang jauh lebih mudah ditebak kuncinya dibandingkan dengan RSA.

Jika dilihat dari kecepatan proses pembangkitan kunci, dari kedua algoritma tersebut, yang lebih cepat dalam pembangkitan kunci adalah algoritma Rabin. Pembangkitan kunci pada algoritma RSA lebih lama dibandingkan algoritma Rabin karena menggunakan 3 buah nilai yang berbeda.

Kunci publik yang dihasilkan oleh algoritma RSA terdiri dari dua buah nilai dan algoritma Rabin terdiri dari satu nilai. Kunci privatnya dari keduanya sama-sama terdiri dari dua nilai. Melihat banyaknya nilai yang dihasilkan, maka penyimpanan kunci publik dan privat dibutuhkan oleh algoritma RSA meskipun pada kenyataannya kedua algoritma tersebut membutuhkan penyimpanan kunci publik dan privat karena angka yang dibentuk dari pembangkitan kunci bukan integer biasa, tetapi mungkin angka tersebut merupakan angka 256 bit atau 512 bit yang sulit untuk dihapalkan.

Jika dibandingkan, proses enkripsi metode Rabin merupakan proses yang lebih sederhana dibandingkan dengan RSA. Kecepatan proses enkripsi Rabin jauh lebih baik dibandingkan dengan RSA karena menggunakan komputasi yang sederhana. Selain itu juga untuk sumber daya yang terbatas, Rabin lebih unggul.

Hasil dari enkripsi RSA dan Rabin sama-sama baik dan memiliki tingkat keamanan yang tinggi walaupun tidak setinggi hasil enkripsi algoritma. Karena hasil enkripsi dari kedua metode tersebut hampir tidak dapat dibedakan satu sama lain, dapat dikatakan bahwa hasil enkripsi dari kedua algoritma tersebut memiliki tingkat keamanan yang setingkat.

Tingkat keamanan kedua teknik ini relatif sama karena keduanya mengandalkan kelebihan sulitnya memfaktorkan bilangan yang berukuran besar. Disamping itu, mengingat bahwa teknik Rabin merupakan varian dari teknik RSA dapat dikatakan tingkat keamanan keduanya relatif sama.

Proses dekripsi pada RSA membutuhkan waktu sedikit

lebih lama dibandingkan dengan metode Rabin. Akan tetapi, teknik Rabin terbukti memiliki kekurangan pada proses dekripsinya karena dia melakukan proses yang kurang efektif, yaitu melakukan proses Chinese remainder sebanyak empat kali, sementara hasil dekripsi yang benar hanya satu dan penerima harus menentukan sendiri mana yang benar.

Hasil dekripsi dari RSA sangat akurat dengan plainteksnya. Kemungkinan kesalahan yang terjadi sangat kecil. Namun berbeda dengan metode Rabin, dimana dia menghasilkan empat kemungkinan plainteks dimana penerima pesan harus dapat menentukan sendiri plainteks sebenarnya diantara keempat kemungkinan yang dihasilkan oleh proses dekripsi.

V. Kesimpulan

Melihat dari perbandingan yang sudah dilakukan, maka kesimpulan yang dapat diambil adalah :

- Rabin merupakan salah satu varian dari RSA
- Rabin memiliki proses komputasi yang lebih sederhana dibandingkan dengan RSA
- Hasil dekripsi Rabin menghasilkan 4 kemungkinan plainteks yang benar
- Tingkat keamanan kedua algoritma relatif sama.

Referensi

[1] Munir, Rinaldi, Slide Kuliah IF3058, Kriptografi, bagian Kriptografi Kunci-Publik, 2012.

[2] Munir, Rinaldi, Slide Kuliah IF3058, Kriptografi, bagian Algoritma RSA, 2012.

[3] Cormen, Thomas H.; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein (2001). Introduction to Algorithms (2e ed.). MIT Press and McGraw-Hill.

[4] Rabin, Michael. Digitalized Signatures and Public-Key Functions as Intractable as Factorization. MIT Laboratory for Computer Science, January 1979.

[5] Christof Paar, Jan Pelzl, "Introduction to Public-Key Cryptography", Springer, 2009 .

[6] IEEE 1363: Standard Specifications for Public-Key Cryptography

[7] Pinkas, Benny, Rabin's Encryption Systems, Digital Signature, 2005.

[8] Rădulescu, Mihnea, Public-Key Cryptography : the RSA and the Rabin Cryptosystems, 2008.

[9] Buchmann, Johannes. *Einführung in die Kryptographie*. Second Edition. Berlin: Springer, 2001.

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 Mei 2012

Tadya Rahanady H (13509070)