IMPLEMENTASI *CHINESE REMAINDER THEOREM* DALAM MEMBENTUK VARIAN RSA (RIVEST-SHAMIR-ADLEMAN) UNTUK PENGAMANAN DATA DIGITAL

Putri Erivani – NIM 13505033

Program Studi Teknik Informatika,Sekolah Teknik Elektro dan Informatika,
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if115033@students.if.itb.ac.id

ABSTRAK

Saat ini, banyak cara yang digunakan untuk melindungi data digital yang disimpan ataupun dikirim melalui media elektronik. Salah satunya adalah RSA(*Rivest-Shamir-Adleman*). RSA standar menggunakan aritmatika modular untuk melakukan proses enkripsi dan dekripsinya.

Makalah ini membahas tentang *Chinese Remainder Theorem* , RSA(*Rivest-Shamir-Adleman*), dan implementasi *Chinese Remainder Theorem* untuk membentuk varian RSA, yaitu RSA-CRT dan RSA-CRT seimbang (*rebalanced RSA-CRT*) yang proses dekripsinya dapat mencapai tiga kali lebih cepat daripada RSA standar yang menggunakan aritmatika modular.

Kata kunci : *Chinese Remainder Theorem*, RSA, RSA-CRT, *rebalanced RSA-CRT*, enkripsi, dekripsi, aritmatika modular

1. PENDAHULUAN

Teknologi komunikasi elektronik telah berkembang dengan sangat cepat dalam beberapa dekade terakhir, menciptakan aplikasi-aplikasi dan kesempatan-kesempatan baru sepanjang jalannya. Saat ini, kita dapat mengirim atau menerima pesan multimedia dari seseorang yang “semu” di seluruh dunia melalui internet. Agar data yang dikirimkan tidak disadap atau dicuridengar oleh orang selain orang yang dituju, kita perlu menyamarkan pesan sebelum mengirimnya melalui saluran komunikasi yang tidak aman.

Hal ini dapat dilakukan dengan *cryptosystem*. Awalnya, *cryptosystem* menggunakan kunci yang simetris. Simetris maksudnya algoritma ini membutuhkan kunci rahasia yang sama untuk enkripsi dan dekripsi dan algoritma enkripsi dan dekripsi pada dasarnya sama. Algoritma dengan kunci simetris ini dapat dianalogikan sebagai kotak penyimpanan dengan kunci yang kuat [1]. Setiap orang yang memiliki kunci dapat menyimpan pesan di dalamnya dan menerima pesan.

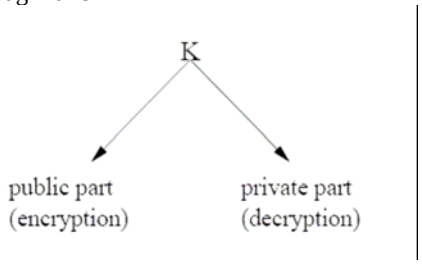
Masalah utama dengan skema kunci simetris adalah :

1. membutuhkan perpindahan kunci rahasia yang aman
2. dalam lingkungan jaringan (*network*), setiap pasang pengguna harus mempunyai kunci yang berbeda yang menyebabkan terlalu banyak kunci

Ide baru :

Membuat celah pada kotak penyimpanan sehingga semua orang dapat memasukkan pesan, namun hanya penerima yang dapat membuka kotak dan melihat isinya.

Ide : bagi kunci



Gambar 1. Ide membagi kunci

Pada tahun 1978, Ron Rivest, Adi Shamir, dan Len Adleman menemukan suatu cara untuk mengimplementasikan *cryptosystem* dengan kunci publik, yang dikenal dengan RSA *cryptosystem*. Metode ini menyediakan keamanan tingkat tinggi dan mudah diimplementasikan, sehingga dalam waktu yang singkat metode ini menjadi *cryptosystem* dengan kunci publik yang paling banyak digunakan.

Dalam RSA, baik enkripsi maupun dekripsi merupakan pemangkatan modular (*modular exponentiation*) yang dapat dilakukan dengan serangkaian perkalian modular. Proses ini memerlukan waktu komputasi yang relatif lama. Salah satu cara untuk mengurangi waktu komputasi adalah menggunakan C*hinese Remainder Theorem (CRT)*, karena C*hinese Remainder Theorem* diketahui dapat mereduksi waktu komputasi RSA dengan metode *pecahbelah-lalu-kuasai(divide-and-conquer method)*

Dengan mengambil keuntungan dari *Chinese Remainder Theorem* (CRT), usaha yang digunakan untuk mengkomputasi dekripsi RSA dapat direduksi secara signifikan.

Jika kedua bilangan prima p dan q yang membangun modulo N diketahui, adalah mungkin menghitung pemangkatan modular $M = C^D \bmod N$ secara terpisah mod p dan mod q dengan pangkat yang lebih pendek. Karena panjang dari bilangan pangkat adalah sekitar $n/2$, kira-kira $3n/4$ perkalian modular diperlukan untuk setiap pemangkatan modular[2].

2. CRT (*CHINESE REMAINDER THEOREM*)

C*hinese Remainder Theorem* pertama kali dikenal pada abad pertama. Teorema ini disebut "*Chinese*" karena contoh numeriknya tercatat dalam manuskrip China pada tahun 300 M (dikarang oleh Sun Tse), dan kasus umumnya dicatat dan dibuktikan oleh Ch'in Chiu-Shao pada tahun 1247 M.

C*hinese Remainder Theorem* banyak diaplikasikan dalam ilmu komputer. Beberapa diantaranya adalah algoritma dekripsi RSA, algoritma logaritmik diskrit, algoritma untuk me-*recover* rahasia dalam skema *Mignotte's threshold secret sharing* atau skema *Asmuth-Bloom threshold secret sharing.*[4]

Teorema 1 (*Chinese Remainder Theorem*)

Misalkan $m_1, m_2, ...,m_n$ adalah bilangan bulat positif sedemikian sehingga FPB(m_i, m_j)=1 untuk $i \neq j$. Maka sistem kongruen lanjar

$$x \equiv a_k \pmod{m_k}$$

mempunyai sebuah solusi unik modulo $m=m_1 \cdot m_2 \cdot ... \cdot m_n$.[5]

Contoh : Cari bilangan integer x yang memenuhi kriteria : menyisakan 3 ketika dibagi dengan 5, menyisakan 5 jika dibagi dengan 7, dan menyisakan 7 ketika dibagi dengan 11.

Dengan kata lain, x harus memenuhi kekongruenan berikut :

$$x \equiv 3 \pmod 5$$
$$x \equiv 5 \pmod 7$$
$$x \equiv 7 \pmod{11}$$

Bilangan modulus dapat berupa bilangan apa saja (dalam hal ini 5,7,dan 11), tetapi tidak ada pasangan dua bilangan modulus yang mempunyai nilai faktor pembilang terbesar (*great common divisor*) lebih dari 1.

Untuk menyelesaikan masalah, ambil kekongruenan pertama, $x \equiv 3 \pmod 5$ ekuivalen dengan $x = 3 + 5k_1$ untuk beberapa nilai k. Substitusikan ini ke kongruen kedua menjadi $3 + 5k_1 \equiv 5 \pmod 7$, dari persamaan ini diperoleh $k_1 \equiv 6 \pmod 7$, atau $k_1 = 6 + 7k_2$ untuk beberapa nilai k_2. Jadi kita mendapatkan $x = 3 + 5k_1 = 3 + 5(6 + 7k_2) = 33 + 35k_2$ yang memenuhi dua kongruen pertama. Jika x memenuhi kongruen yang ketiga, kita harus mempunyai $33 + 35k_2 \equiv 7 \pmod{11}$, yang menyebabkan $k_2 \equiv 9 \pmod{11}$ atau $k_2 = 9 +$

$11k_3$. Substitusi k_2 ini ke dalam kongruen yang ketiga menghasilkan $x = 33 + 35(9 + 11k_3) \equiv 348 + 385k_3 \pmod{11}$. Dengan demikian, $x \equiv 348 \pmod{385}$ yang memenuhi ketiga kongruen tersebut. Dengan kata lain, 348 adalah solusi unik modulo 385. dapat dilihat bahwa $385 = 5 \cdot 7 \cdot 11$.

Solusi unik ini mudah dibuktikan sebagai berikut. Solusi tersebut modulo

$$\begin{aligned} m &= m_1 \cdot m_2 \cdot m_3 \\ &= 5 \cdot 7 \cdot 11 \\ &= 5 \cdot 77 \\ &= 11 \cdot 35. \end{aligned}$$

Karena $77\ \ 3 \equiv 1 \pmod{5}$, $55 \cdot 6 \equiv 1 \pmod{7}$, dan $35 \cdot 6 \equiv 1 \pmod{11}$, solusi unik dari sistem kongruen tersebut adalah

$$\begin{aligned} x &\equiv 3 \cdot 77 \cdot 3 + 5 \cdot 55 \cdot 6 \ + 7 \cdot 35 \cdot 6 \pmod{385} \\ &\equiv 3813 \pmod{385} \\ &\equiv 348 \pmod{385} \end{aligned}$$

Chinese Remainder Theorem mempunyai banyak kegunaan, misalnya :

- *Chinese Remainder Theorem* digunakan untuk membagi cincin/lingkaran besar menjadi banyak cincin/lingkaran kecil yang membantu membuktikan hal-hal tertentu dengan cara yang jauh lebih baik karena cincin/lingkaran yang lebih kecil lebih mudah ditangani daripada cincin/lingkaran yang lebih besar.
- *Chinese Remainder Theorem* juga digunakan untuk mendesain algoritma baru dan mempercepat algoritma yang telah ada beberapa kali [7]

3. RSA (RIVEST-SHAMIR-ADLEMAN)

RSA adalah akronim dari nama para penemunya, yaitu Ron Rivest, Adi Shamir, dan Len Adleman. Metode ini pertama kali dipublikasikan dalam *Scientific American.* Algoritma RSA termasuk bagian dari *web browser* dari Microsoft dan Netscape dan digunakan oleh SSL (*Secure Socket Layer*) yang menjamin keamanan dan privasi di internet. Metode ini berdasarkan pada ide : "mengalikan dua bilangan adalah mudah, khususnya dengan komputer. Namun memfaktorkan bilangan dapat menjadi sangat sulit". Sebagai contoh, adalah relatif mudah untuk mengambil dua bilangan prima p dan q dan menghitung hasil kalinya $N = pq$. Namun jika diberikan nilai N, sulit untuk menemukan faktornya p dan q, khususnya untuk bilangan N yang besar. Enkripsi menggunakan nilai atau kunci publik (*public key*), yang disebarluaskan dan diketahui semua orang yang ingin mengirim pesan. Sedangkan dekripsinya menggunakan sebuah kunci pribadi (*private key*) yang dijaga kerahasiannya oleh penerima dan tidak dapat dideduksi dari kunci publik. Kriptografi dengan kunci publik bekerja tanpa mengharuskan kedua pihak menjaga kerahasiaan, kunci pribadi tidak pernah perlu diberitahu ke pengirim pesan.

3.1. Cara Kerja RSA

Kriptografi dengan RSA bekerja berdasarkan teorema berikut:

Teorema 1 (Fermat's Little Theorem)

Jika p adalah bilangan prima, dan a adalah sebuah integer sedemikian hingga FPB(a,p)=1, maka

$$a^{p-1} \equiv 1 \pmod{p}$$

Bukti : Misalkan bilangan $(a \cdot 1), (a \cdot 2), ..., (a \cdot (p-1))$, semuanya modulo p. Semua bilangan tersebut berbeda. Jika ada diantara bilangan tersebut yang sama, misalkan $a \cdot m = a \cdot n \pmod{p}$, maka $a \cdot (m-n) = 0 \pmod{p}$ sehingga $m - n$ merupakan kelipatan dari p. Namun karena semua m dan n kurang dari p, $m=n$.

Sebagai hasilnya $(a \cdot 1), (a \cdot 2), ..., (a \cdot (p-1))$ haruslah merupakan susunan ulang dari $1,2,...,(p-1)$. Sehingga kita mempunyai modulo p :

$$\prod_{i=1}^{p-1} i = \prod_{i=1}^{p-1} a \cdot i = a^{p-1} \prod_{i=1}^{p-1} i,$$

sehingga $a^{p-1} = 1 \pmod{p}$.

Teorema 2 (Fermat's Theorem Extension)

Jika FPB(a,m) = 1 maka $a^{\phi(m)}=1 \pmod{m}$*,dimana* $\phi(m)$ *adalah jumlah bilangan integer kurang dari m yang relatif prima terhadap m. Bilangan m tidak harus prima*.

Bukti : misalkan $\phi(m) = n$. Kemudian misalkan *n* bilangan kurang dari *m* yang relatif prima terhadap *m* adalah :

$$a_1 , a_2 , a_3 , \ldots , a_n$$

Maka $a \cdot a_1 , a \cdot a_2 , a \cdot a_3 , \ldots , a \cdot a_n$ juga adalah relatif prima terhadap *m*, dan harus semuanya berbeda, sehingga bilangan-bilangan tersebut haruslah merupakan susunan ulang dari $a_1 , a_2 , a_3 , \ldots , a_n$.

Sehingga :

$$\prod_{i=1}^{n} a_i = \prod_{i=1}^{n} a \cdot a_i = a^n \prod_{i=1}^{n} a_i,$$

modulo *m*, sehingga $a^n = 1 \pmod{m}$.

3.2. Algoritma RSA

1. Pilih dua buah bilangan prima sembarang, sebut *a* dan *b*. Jaga kerahasiaan *a* dan *b* ini.
2. Hitung $n = a \times b$. Besaran n tidak dirahasiakan.
3. Hitung $m = (a - 1) \times (b - 1)$. Sekali *m* telah dihitung, *a* dan *b* dapat dihapus untuk mencegah diketahuinya oleh pihak lain.
4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya *e*, yang relatif prima terhadap *m*.
5. Bangkitkan kunci dekripsi, *d*, dengan kekongruenan $ed \equiv 1 \pmod{m}$. Lakukan enkripsi terhadap isi pesan dengan persamaan $c_i = p_i^e \bmod n$, yang dalam hal ini p_i adalah blok plainteks, c_i adalah chiperteks yang diperoleh, dan *e* adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa nilai p_i harus terletak dalam himpunan nilai 0, 1, 2, …, $n - 1$ untuk menjamin hasil perhitungan tidak berada di luar himpunan.
6. Proses dekripsi dilakukan dengan menggunakan persamaan $p_i = c_i^d \bmod n$, yang dalam hal ini *d* adalah kunci dekripsi. [5]

Contoh : Misalkan $a = 47$ dan $b = 71$ (keduanya prima), maka dapat dihitung

$n = a \times b = 3337$ dan
$m = (a - 1) \times (b - 1) = 3220$.

Pilih kunci publik $e = 79$ (yang relatif prima dengan 3220 karena pembagi bersama terbesarnya adalah 1). Nilai *e* dan *m* dapat dipublikasikan ke umum.

Selanjutnya akan dihitung kunci dekripsi *d* seperti yang dituliskan pada langkah instruksi 4,

$$e \times d \equiv 1 \pmod{m}$$

Kunci dekripsi *d* sebagai berikut:

$$d = \frac{1 + (k \times 3220)}{79}$$

Dengan mencoba nilai-nilai $k = 1, 2, 3, \ldots$, diperoleh nilai *d* yang bulat adalah 1019. Ini adalah kunci dekripsi.

Misalkan plainteks

P = HARI INI

atau dalam desimal ASCII:

7265827332737873

Pecah *P* menjadi blok yang lebih kecil (misal 3 digit):

$p_1 = 726$ $\quad$ $p_4 = 273$
$p_2 = 582$ $\quad$ $p_5 = 787$

$p_3 = 733$ $p_6 = 003$

Blok pertama dienkripsikan sebagai $726^{79} \bmod 3337 = 215 = c_1$.

Blok kedua dienkripsikan sebagai $582^{79} \bmod 3337 = 776 = c_2$.

Dengan melakukan proses yang sama untuk sisa blok lainnya, dihasilkan chiperteks C = 215 776 1743 933 1731 158.

Proses dekripsi dilakukan dengan menggunakan kunci rahasia d = 1019.

Blok c_1 didekripsikan sebagai $215^{1019} \bmod 3337 = 726 = p_1$,

Blok c_2 didekripsikan sebagai $776^{1019} \bmod 3337 = 582 = p_2$.

Blok plainteks yang lain dikembalikan dengan cara yang serupa. Akhirnya kita memperoleh kembali plainteks semula

P = 7265827332737873

yang karakternya adalah

P = HARI INI.

Perhitungan perpangkatan pada proses enkripsi ($c_i = p_i^e \bmod n$)dan dekripsi ($p_i = c_i^d \bmod n$) membutuhkan bilangan yang sangat besar. Untuk menghindari penggunaan bilangan yang besar, maka dapat digunakan penyederhanaan dengan persamaan berikut:

$$ab \bmod m = [(a \bmod m)(b \bmod m)] \bmod m$$

3.3. Kekuatan dan Keamanan RSA

- Kekuatan algoritma *RSA* terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya, yang dalam hal ini $n = a \times b$.
- Sekali n berhasil difaktorkan menjadi a dan b, maka $m = (a - 1)\times(b - 1)$ dapat dihitung. Selanjutnya, karena kunci enkrispi e diumumkan (tidak rahasia), maka kunci dekripsi d dapat dihitung dari persamaan $e \times d \equiv 1 \pmod m$. Ini berarti proses dekripsi dapat dilakukan oleh orang yang tidak berhak.
- 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. Bayangkanlah berapa besar usaha kerja yang diperlukan untuk memfaktorkan bilangan bulat 200 digit menjadi faktor primanya. Menurut Rivest dan kawan-kawan, usaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun! (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik).[5]

3.4. Serangan Terhadap RSA

Ada beberapa serangan terhadap implementasi RSA. Serangan-serangan ini umumnya mengeksploitasi kelemahan dalam cara RSA digunakan, bukan merusak algoritma RSA. Berikut ini adalah list serangan terhadap algoritma RSA yang mungkin ,secara teori, dapat dilakukan[1] :

1. *Brute force*
   Diberikan $y = x^e \bmod n$, coba semua nilai kunci d; $0 \le e < \phi(N)$ untuk memenuhi $x = y^d \bmod n$. Dalam prakteknya $|K| = \phi(N) \approx N > 2^{500} \Rightarrow$ tidak mungkin
2. Mencari $\phi(N)$
   Diberikan n, b, $y = x^b \bmod n$, cari $\phi(N)$ dan hitung $a = b^{-1} \bmod \phi(N)$.
   $\Rightarrow$ menghitung $\phi(N)$ dipercaya sesulit memfaktorkan N.
3. Langsung mencari nilai a
   Diberikan n, b, $y = x^b \bmod n$, olangsung cari nilai a dan hitung $x = y^a \bmod n$
   $\Rightarrow$ menghitung a dipercaya sesulit memfaktorkan N.
4. Memfaktorkan N

Diberikan n, b, $y = x^b \bmod n$, faktor $p \cdot q = n$ lalu hitung
$\phi(N) = (p\text{-}1)(q\text{-}1)$
$b = a^{-1} \bmod \phi(N)$
$x = y^a \bmod N$

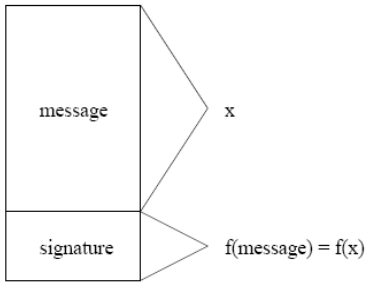
3.5. Penggunaan RSA

Algoritma RSA dapat digunakan dalam :

1. *electronic mail (e-mail)*
2. *electronic funds transfer*
3. *electronic data interchange*
4. distribusi perangkat lunak
5. penyimpanan data
6. aplikasi yang membutuhkan jaminan integritas data dan autentikasi data asli
7. *digital signature* dan *signature verification*

3.6. Digital Signature dan Signature Verification

Prinsipnya sama dengan *signature* konvensional di kertas. Diberikan suatu pesan *x*, sebuah *digital signature* ditambahkan ke pesan tersebut. Sama seperti *signature* konvensional, hanya orang yang mengirim pesan yang dapat membuat *signature* yang sah. Bertujuan untuk mencapai hal ini dengan kriptografi, kita dapat membuat *signature* sebagai sebuah fungsi dari kunci rahasia, sehingga hanya pemegang kunci rahasia yang dapat menandai pesan. Untuk meyakinkan bahwa *signature* berubah pada tiap dokumen, kita juga membuat *signature* sebagai fungsi dari pesan yang ditandai.



Gambar 2. *digital signature* dan blok pesan

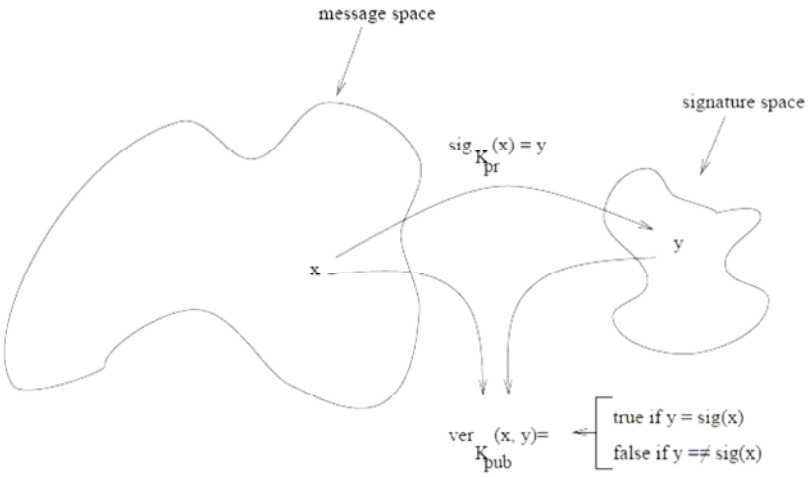
Untuk menandai dokumen, pengirim dapat melakukan hal berikut :

1. Buat pesan yang akan dikirim
2. Representasikan pesan ini sebagai integer *m* antara 0 dan $n-1$.
3. Gunakan kunci rahasia (n,d) untuk menghitung *signature* $s = m^d \bmod n$.
4. Kirim *signature* *s* ini kepada penerima

Untuk verifikasi *signature* penerima dapat melakukan hal berikut:

1. Gunakan kunci publik si pengirim (n,e) untuk menghitung integer $v = s^e \bmod n$.
2. Pisahkan isi pesan dari integer ini.
3. Secara terpisah, hitung isi pesan dari informasi yang telah ditandai
4. Jika kedua isi pesan sama/identik, maka *signature* valid

Keuntungan utama yang dimiliki oleh *digital signature* adalah memungkinkan pihak-pihak yang berkomunikasi untuk membuktikan bahwa yang membuat pesan benar-benar salah satu pihak dari mereka. Pembuktian ini bahkan dapat berarti kelegalan.



Gambar 3. *digital signature* dan *message domain*

Kompleksitas dekripsi RSA $M = C^D \bmod N$ sangat bergantung pada nilai D dan N. Perpangkatan dekripsi D menunjukkan jumlah perkalian modular yang harus perlu dilakukan untuk melakukan perpangkatan, dan modulo N menentukan besar hasil intermediat. Salah satu cara memperkecil nilai D dan N adalah dengan menggunakan *Chinese Remainder Theorem(CRT)*[2]

4. *CHINESE REMAINDER THEOREM* DALAM RSA-CRT DAN *REBALANCED RSA-CRT*

Dalam RSA-CRT, adalah cara yang umum untuk menggunakan *Chinese Remainder Theorem* selama dekripsi. Pemakaian *Chinese Remainder Theorem* menghasilkan dekripsi yang jauh lebih cepat daripada RSA standar yang menggunakan pemangkatan modular. RSA-CRT berbeda dari RSA standar dalam hal pembangkitan kunci dan dekripsi. Nilai pangkat rahasia d tidak dapat dibuat pendek.

Jika $d < N^{0.292}$, sistem RSA dapat dihancurkan secara menyeluruh.[8]

4.1. Membangkitkan kunci RSA-CRT

1. Misalkan p dan q adalah dua bilangan prima yang sangat besar dengan ukuran yang hampir sama sedemikian hingga FPB(p-1, q-1) = 2.
2. Hitung $N = p*q$.
3. Ambil dua bilangan integer secara acak d_p dan d_q sedemikian sehingga FPB(d_p, p-1) = 1, FPB(d_q, q-1) = 1, dan $d_p \equiv d_q \pmod 2$.
4. Cari bilangan d sedemikian sehingga $d \equiv d_p \pmod{p\text{-}1}$ dan $d \equiv d_q \pmod{q\text{-}1}$.
5. Hitung $e = d^{-1} \pmod{\phi(N)}$

Kunci publik adalah $<N,e>$ dan kunci rahasia adalah $<p,q,d_p,d_q>$. Karena FPB(d_p, p-1) = 1 dan $d \equiv d_p \pmod{p\text{-}1}$, kita mempunyai FPB($d$,$p$-1) = 1. Dengan cara yang sama, F PB(d,q-1) = 1. Sebagai akibatnya FPB(d,$\phi(N)$)

= 1, dan karena langkah 5, *e* dapat dihitung nilainya.

Untuk mengaplikasikan *Chinese Remainder Theorem* pada langkah 4, bilangan modulo masing-masing (dalam hal ini *p*-1 dan *q*-1) harus pasangan bilangan yang relatif prima agar persoalan ini mempunyai solusi. Kita perhatikan bahwa *p*-1 dan *q*-1 adalah bilangan genap dan karenanya kita tidak dapat langsung mengaplikasikan *Chinese Remainder Theorem*. Bagaimanapun, FPB((*p*-1)/2, (*q*-1)/2) = 1. Karena FPB(d_p, *p*-1) = 1 dan FPB(d_q, *q*-1) = 1, didapatkan d_p, d_q adalah bilangan integer ganjil dan *dp*-1, *dq*-1 adalah bilangan integer genap. Kita punya FPB(*d*,*p*-1) = 1, yang menunjukkan bahwa *d* adalah bilangan ganjil dan *d*-1 adalah bilangan genap.

Untuk memperoleh solusi

$$d \equiv dp \pmod{p\text{-}1},$$
$$d \equiv dq \pmod{q\text{-}1}$$

kita mencari solusi dari

$$d\text{-}1 \equiv dp - 1 \pmod{p\text{-}1},$$
$$d\text{-}1 \equiv dq - 1 \pmod{q\text{-}1}.$$

Dengan menggunakan hukum kanselasi (*cancellation law*) dan menarik faktor 2 keluar, kita mempunyai

$$x=d' \equiv (d\text{-}1)/2==(dp-1)/2 \pmod{(p\text{-}1)/2},$$
$$x=d' \equiv (d\text{-}1)/2==(dq-1)/2 \pmod{(q\text{-}1)/2}.$$

Dengan menggunakan *Chinese Remainder Theorem* didapatkan nilai *d* sedemikian sehingga *d* = (2 * *d'*)+1.

4.2. Dekripsi RSA-CRT

Karena enkripsi RSA-CRT sama dengan prosedur enkripsi RSA standar [8], saat ini perhatian difokuskan pada dekripsi RSA-CRT.

Misalkan *M* adalah *plaintext* dan *C* adalah *ciphertext.*

Teorema :

Jika C tidak habis dibagi oleh p dan $d_p \equiv d \pmod{p-1}$, *maka* $C^{dp} \equiv C^d \pmod{p}$.

Untuk dekripsi kita temukan

1. $Mp = Cdp \pmod{p} = Cd \pmod{p}$ dan $Mq = Cdq \pmod{q} = Cd \pmod{q}$.
2. Kemudian dengan menggunakan *Chinese Remainder Theorem,* didapatkan solusi untuk

$$M = M_P \pmod{p}$$
$$= C^d \pmod{p},$$
$$M = Mq = C^{dq} \pmod{q}$$
$$= C^d \pmod{q}.$$

Contoh : Pilih *p* = 7, *q* = 11,
FPB(*p*-1, *q*-1) = 2, *N* = *p***q* = 7*11 = 77,
$\phi(N) = (p\text{-}1)*(q\text{-}1) = 6*10 = 60$.
Misalkan d_p = 5, FPB(d_p , *p*-1) = FPB(5,6) = 1.
d_q = 3, FPB(d_q , *q*-1) = FPB(3,10) = 1.

Kita akan mencari nilai *d* sedemikian sehingga

$$d \equiv 5 \pmod{6},$$
$$d \equiv 3 \pmod{10}.$$

Kita tidak dapat langsung menggunakan *Chinese Remainder Theorem* karena FPB(6,10) ≠ 1, oleh karena itu kita mengubah sistem kekongruenan sedemikian sehingga hukum kanselasi (*cancellation law*) dapat diaplikasikan.

Oleh karena itu, kita mempunyai

$$d\text{-}1 \equiv 5\text{-}1 \bmod 6,$$
$$d\text{-}1 \equiv 3\text{-}1 \bmod 10.$$

Dengan menggunakan *cancellation law*

$$(d\text{-}1)/2 \equiv (5\text{-}1)/2 \pmod{(6/2)},$$
$$(d\text{-}1)/2 \equiv (3\text{-}1)/2 \pmod{(10/2)}.$$
$$x = d' = (d\text{-}1)/2 \equiv 2 \bmod 3,$$
$$x = d' = (d\text{-}1)/2 \equiv 1 \bmod 5.$$

Selesaikan dengan menggunakan *Chinese Remainder Theorem,*

$M = 3*5 = 15,$
$M1 = 15/3 = 5,$
$M2 = 15/5 = 3.$
$5*N1 \equiv 1 \bmod 3,\ N1 = 2,$
$3*N2 \equiv 1 \bmod 5,\ N2 = 2.$

Kita punya ,

$$d' = x = 2*5*2 + 1*3*2 = 26 (\bmod 15) = 11.$$

Oleh karena itu

$d' = 11$ dan
$d = (2*d')+1 = (2*11) +1$
$= 23$

Sekarang kita mencari nilai e sedemikian sehingga

$e*d \equiv 1 \bmod \phi(N),$
$e*23 \equiv 1 \bmod 60,\ e = 47$

Misalkan *plaintext* $M = 5$.

$$C = 5^{47} (\bmod 77) = 3$$

Untuk dekripsi kita dapatkan

$M = M_p \ (\bmod p) = c^d \ (\bmod p),$
$M = M_q \ (\bmod q) = c^d \ (\bmod q).$
$M_p = 35 \ (\bmod 7) = 243 \ (\bmod 7)$
$= 5,$
$M_q = 33 \ (\bmod 11) = 27 \ (\bmod 11)$
$= 5.$

Dengan menggunakan *Chinese Remainder Theorem,*

$M = 7*11 = 77,$
$M1 = 77/7 = 11,$
$M2 = 77/11 = 7.$
$11*N1 \equiv 1 \bmod 7,\ N1 = 2,$
$7*N2 \equiv 1 \bmod 11,\ N2 = 8.$
$x = 5*11*2 + 5*7*8 = 390 \bmod 77 = 5$

Didapat $x = M = 5$, seperi yang diharapkan. Dalam contoh spesifik ini (M_p dan M_q) = 5 adalah solusi umum dan tidak perlu mengaplikasikan *Chinese Remainder Theorem* lebih jauh.

4.3. *Rebalanced* RSA-CRT

Sekarang perhatian kita alihkan ke varian RSA yang lain, yaitu *Rebalanced* RSA-CRT. Tujuan utama dari *Rebalanced* RSA-CRT adalah mempercepat dekripsi RSA dengan menggilirkan tugas kepada enkripter. Sifat ini bermanfaat untuk dekripsi RSA di perangkat bergerak (*mobile devices*) seperi telepon selular yang hidupnya dibatasi oleh kemampuan baterai.

Dekripsi r*ebalanced* RSA-CRT dapat mencapai lebih dari tiga kali lebih cepat daripada RSA standar. Satu-satunya perbedaan antara RSA-CRT dengan *Rebalanced* RSA-CRT adalah dalam memilih nilai d_p *dan* d_q. dalam *Rebalanced* RSA-CRT, ukuran nilai e dan d adalah bagian dari $\phi(N)$, dimana dalam RSA standar, e biasanya bilangan integer dengan panjang 16-bit atau 32-bit.

Menurut [9], ukuran d_p dan d_q seharusnya minimal 160-bit untuk mencapai keamanan tingkat 2^{80}. Sebagau hasilnya, untuk *Rebalanced* RSA-CRT selalu dipilih (d_p dan d_q) > 160 bit. Langkah-langkah seterusnya sama dengan langkah-langkah pada RSA-CRT.

Kelemahan utama dari skema ini adalah tugas enkripter sangant besar, bahkan untuk suatu komputer *high-end* sekalipun.

5. KESIMPULAN

Kesimpulan yang dapat diambil adalah :

1. Pada RSA standar, baik enkripsi maupun dekripsi merupakan pemangkatan modular (*modular exponentiation*) yang dapat dilakukan dengan serangkaian perkalian modular. Proses ini memerlukan waktu komputasi yang relatif lama
2. Ada beberapa serangan yang secara teori dapat dilakukan terhadap implementasi RSA, yaitu *brute force*, mencari nilai $\phi(N)$, langsung mencari kunci rahasia *d*, dan dengan emfaktorkan *n*.
3. Waktu komputasi dapat direduksi dengan mengimplementasikan C*hinese Remainder Theorem (CRT*) pada RSA.
4. RSA-CRT berbeda dari RSA standar dalam hal pembangkitan kunci dan dekripsi.
5. Dekripsi r*ebalanced* RSA-CRT dapat mencapai lebih dari tiga kali lebih cepat daripada RSA standar.
6. *Rebalanced* RSA-CRT mempercepat dekripsi RSA dengan cara menggilirkan tugas kepada enkripter. Namun, penggiliran tugas tersebut berakibat pada besarnya beban yang ditanggung enkripter.
7. Perbedaan antara RSA-CRT dengan *Rebalanced* RSA-CRT adalah dalam memilih nilai d_p *dan* d_q.

6. DAFTAR PUSTAKA

[1] Paar , Christof. 2005. *Applied Cryptography And Data Security*. http://www.crypto.ruhr-uni-bochum.de/imperia/md/content/lectures/notes.pdf
waktu akses: 30 Desember 2006 pukul 12:00

[2] Großsch¨adl, Johann.*The Chinese Remainder Theorem and its Application in a High-Speed RSA Crypto Chip* *http://www.acsac.org/2000/papers/48.pdf*
waktu akses: 30 Desember 2006 pukul 11:55

[3] Davis, Tom. 2003. *RSA Encryption* http://www.geometer.org/mathcircles/RSA.pdf
waktu akses: 30 Desember 2006 pukul 12:13

[4] Iftene, Sorin. *Compartmented Secret Sharing Based on the Chinese Remainder Theorem* *http://eprint.iacr.org/2005/408.pdf*
waktu akses: 30 Desember 2006 pukul 12:30

[5] Munir, Rinaldi. 2004. *Bahan Kuliah IF2152 Matematika Diskrit.* Departemen Teknik Informatika, Institut Teknologi Bandung

[6] Boneh, Dan.1999. *Twenty Years of Attacks on the RSA Cryptosystem* http://www.ams.org/notices/199902/boneh.pdf
waktu akses 2 Januari 2007 12:15

[7] Garg , Rohit. 2004. *Computational Number Theory and Algebra*. http://cobweb.ecn.purdue.edu/~kak/courses-i-teach/compsec/Lecture/Lecture_5.pdf
waktu akses: 30 Desember 2006 pukul 10:23

[8] V., Sarad A. *Applications to Chinese Remainder Theorem* http://neworder.box.sk/files/CRT.pdf
waktu akses 30 Desember 3006 pukul 12:00

[9] Boneh, Dan and Hovav Shacham.2002. *Fast Variants of RSA*. http://www.rsasecurity.com/rsalabs/cryptobytes/CryptoBytes_January_2002_final.pdf
waktu akses: 31 Desember 2006 pukul 16:34

[10] http://en.wikipedia.org/wiki/Chinese_remainder_theorem
waktu akses 30 Desember 2006 pukul 12:05

[11] Lady, E. L. *Chinese Remainder Theorem* http://www.math.hawaii.edu/~lee/courses/Chinese.pdf waktu akses: 2 Januari 2007 pukul 11:50

[12] Wu , Chung-Hsien, Jin-Hua Hong, dan Cheng-Wen Wu. 2001. *RSA Cryptosystem Design Based on the Chinese Remainder Theorem.*