Modifikasi Algoritma RSA dengan *Chinese Reamainder Theorem* dan *Hensel Lifting*

Reyhan Yuanza Pohan [1)]

1) Jurusan Teknik Informatika ITB, Bandung 40132, email: if14126@students.if.itb.ac.id

Abstract *– Masalah pengiriman data ataupun pesan telah menjadi masalah penting pada era teknologi informasi seperti sekarang ini. Terkadang pesan-pesan ini harus bersifat rahasia agar tidak diketahui secara umum. Apabila pesan tersebut diketahui, maka pesan tersebut dapat disalahgunakan untuk kejahatan oleh orang lain.*

Kriptografi merupakan salah satu alat keamanan yang digunakan untuk menyembunyikan suatu pesan. Kriptografi dapat dibedakan menjadi kriptografi kunci-simetri dan kriptografi kunci-nirsimetri atau disebut juga kriptografi kunci-publik. Salah satu metode algoritma kriptografi kunci-publik yang sudah dikenal sejak lama adalah RSA (Rivest-Shamir-Adleman). Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi factor-faktor prima. Namun hal itu juga yang menjadi kekurangan dari algoritma ini, yaitu kecepatan dalam mendekripsi pesan menjadi lambat.

Penulis akan mencoba memodifikasi algoritma RSA menggunakan Chinese Remainder Theorem (CRT) dan Hensel Lifting untuk mempercepat proses dekripsi. CRT dan Hensel Lifting merupakan teorama dalam aritmatika modulo yang akan digunakan dalam melakukan operasi modulo pada proses dekripsi. Dalam makalah ini, penulis akan membahas algoritma RSA, proses enkripsi dan dekripsi sebelum dan sesudah modifikasi. Penulis juga akan melakukan perbandingan kecepatan dan keamanan algoritma sebelum dan sesudah modifikasi.

Kata kunci *: kriptografi, Chinese Remainder Theorem, Hensel Lifting, enkripsi, dekripsi*

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini sudah berkembang secara pesat. Dengan berkembangnya teknologi informasi ini, pertukaran data dapat menjadi suatu permasalahan. Hampir setiap hari pertukaran data terjadi, data-data ini bervariasi besarnya maupun jenisnya. Adakalanya data-data ini bersifat rahasia seperti data pribadi, data organisasi, ataupun data negara. Kerahasiaan ini perlu dijaga agar tidak orang yang menyalahgunakan data-data tersebut. Untuk itulah, kriptografi dikembangkan.

Kriptografi adalah suatu ilmu menyembunyikan informasi. Sampai dengan saat ini, sudah ada berbagai macam algoritma kriptografi, namun secara keseluruhan algoritma kriptografi dibagi menjadi dua yaitu kriptografi kunci-simetri dan kriptografi kunci-nirsimetri atau disebut juga kriptografi kunci-publik. Contoh algoritma kriptografi kunci-publik ini adalah algoritma RSA (Rivest-Shamir-Adleman).

Algoritma RSA sudah dipergunakan secara luas dalam keamanan internet, beberapa contoh penggunaanya adalah untuk tanda tangan digital dan pada komunikasi transaksi. RSA menawarkan keamanan yang baik walaupun membutuhkan komputasi matematika yang kompleks sehingga mengurangi kecepatan dari algoritma ini bila dibandingkan dengan algoritma kunci simetri. Oleh karena itu dibutuhkan suatu metode untuk mempercepat perhitungan yang terdapat pada RSA. Operasi matematika yang terdapat di belakang algoritma ini dapat dibagi menjadi dua operasi, yaitu operasi modular dan perpangkatan. Sehingga untuk membuat implementasi dari RSA yang lebih efisien, maka diperlukan suatu perpangkatan yang efisien di antara dua bilangan modular. Walaupun demikian, perpangkatan modular memiliki kekurangan, yaitu operasi pembagian yang harus dilakukan untuk mendapatkan nilai remainder. Operasi pembagian merupakan operasi yang memakan waktu dan memiliki kompleksitas yang besar.

Solusi dari permasalahan perpangkatan modular adalah dengan mencari suatu metode yang memotong jumlah percobaan operasi pembagian. Banyak algoritma yang sudah didesain untuk menangani masalah ini. Beberapa algoritma ini adalah *Chinese Remainder Theorem* dan *Hensel Lifting*. *Chinese Remainder Theorem* membagi operasi perpangkatan modular yang besar menjadi dua operasi perpangkatan modular yang lebih kecil. Sedangkan *Hensel Lifting* melakukan operasi perpangkatan modular dengan cara memangkatkan nilai modulonya. Kedua algoritma ini akan dijelaskan lebih lanjut pada bab berikutnya.

Dalam makalah ini, penulis akan melakukan modifikasi algoritma RSA dengan *Chinese Remainder Theorem* dan *Hensel Lifting*. Modifikasi ini dilakukan untuk mempercepat proses dekripsi RSA karena proses tersebut memakan waktu lama yang lebih lama dibandingkan proses enkripsi karena perpangkatan modular pada proses dekripsi yang lebih besar.

2. DASAR TEORI

2.1. *RSA*

RSA adalah salah satu dari algoritma kunci publik yang sangat sering digunakan untuk mengotentikasi keaslian suatu data digital. Keamanan enkripsi/dekripsi data dari algoritma kriptografi ini terletak pada kesulitan untuk memfaktorkan modulus n yang sangat besar. Besarnya bilangan yang digunakan mengakibatkan lambatnya operasi yang melibatkan algoritma RSA ini. Dibandingkan dengan algoritma kunci privat seperti DES, RSA membutuhkan waktu komputasi yang lebih lambat pada saat implementasi.

Di tahun 1978, Ron Rivest, Adi Shamir dan Leonard Adleman membuat sebuah algoritma untuk teori penomoran pada sebuah kunci publik, algoritma ini dikenal dengan nama RSA. Pada perkembangannya RSA banyak digunakan karena kemudahannya dan keamanannya.

Sebagai contoh permasalahan, dalam berkomunikasi menggunakan jaringan internet, adanya lubang/celah dalam keamanan menjadi masalah serius karena perkembangan e-mail, e-banking, e-business dan jenis komunikasi lainnya makin pesat untuk mentranfer data-data penting. Sebagai contoh kasus Alice dan Bob yang ingin berkomunikasi dengan privasi tinggi, pada kenyataannya ada pihak ketiga yang dapat mengakses kemunikasi mereka, dalam hal ini misalkan namanya Eve.

Algoritma RSA:

1. Pilih dua bilangan prima sembarang, p dan q.
2. Hitung n = pq (sebaiknya p tidak sama dengan q, sebab juka p sama dengan q maka $n = p^2$ sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n)
3. Hitung $\phi(n) = (p - 1)(q – 1)$.
4. Pilihlah kunci publik, e, yang relatif prima terhadap φ(n). Maksudnya relatif prima adalah bilangan terbesar yang dapat membagi e dan φ(n) untuk menghasilkan nilai 1 (pembagi ini dinyatakan dengan gcd -- greatest common divisor). Algoritma Euclid's digunakan untuk mencari gcd dua bilangan tersebut.
5. Bangkitkan kunci privat dengan menggunakan persamaan ed ≡ 1 (mod φ(n)). Perhatikan bahwa ed ≡ 1 (mod φ(n)) ekivalen dengan ed ≡ 1 + kφ(n), sehingga secara sederhana d dapat dihitung dengan d = (1 + kφ(n)) / e.

Hasil dari algoritma di atas adalah:

- Kunci publik adalah pasangan (e,n)
  - Dipublikasikan bebas
  - Pengiriman balik pesan kepada pemegang kunci privat untuk mengenkripsi pesan
- Kunci privat adalah pasangan (d, n)
  - Rahasia pemegang (end user)
  - Digunakan untuk mendekripsi pesan yang ditujukan kepadanya
  - Dapat berfungsi sebagai digital signature yang beroperasi dengan menggunakan privat key

Catatan: n tidak berifat rahasia, sebab ia diperlukan pada perhitungan enkripsi/dekripsi.

Proses enkripsi dan dekripsi:

- Enkripsi
  1. Ambil kunci publik penerima pesan, e, dan modulus n.
  2. Nyatakan plainteks m menjadi blok-blok m_1, m_2, ..., sedemikian seterusnya sehingga setiap blok merepresentasikan nilai di dalam selang [0, n -1].
  3. Setiap blok m, dienkripsi menjadi blok c_i dengan rumus $c_i = m_i^e \bmod n$.
- Dekripsi
  1. Setiap blok chiperteks c_i didekripsi kembali menjadi blok m_i dengan rumus $m_i = c_i^d \bmod n$.

2.2. *Chinese Remainder Theorem*

Suatu residu, sisa suatu pembagian dari bilangan tertentu yang disebut suatu modulus (yaitu residu 5/7 adalah 2). Bentuk penyajian residu suatu jumlah x yang diatur pada suatu bentuk residu { r_1, r_2,..., r_k} berkenaan dengan modulo { m_1, m_2,…, m_k}.

Theorem 1 (Chinese Remainder Theorem) *Terdapat bilangan-bilangan $n_1,n_2,\ldots,n_k$ adalah bilangan bulat positif di mana relatif prima pada pasangan. Contohnya : PBB(n_i,n_j) = 1 di mana i ≠ j. Lebih jauh lagi, $n = n_1n_2\ldots n_k$ dan $x_1, x_2;\ldots, x_k$ adalah bilangan bulat. Maka sistem kongruen*

$$\begin{aligned} x &\equiv x_1 \bmod n_1 \\ x &\equiv x_2 \bmod n_2 \\ &\vdots \\ x &\equiv x_k \bmod n_k \end{aligned}$$

memiliki solusi yang simultan pada semua kongruen dan dua solusi apapun adalah saling kongruen modulo. Lebih jauh lagi terdapat tepatnya satu solusi antara 0 dan n-1.

Solusi unik dari kongruen simultan memenuhi 0 <= x <= n dapat dihitung dengan :

$$\begin{aligned} x &= \left(\sum_{i=1}^{k} x_i r_i s_i \right) \bmod n \\ &= (x_1 r_1 s_1 + x_2 r_2 s_2 + \ldots + x_k r_k s_k) \bmod n \end{aligned}$$

di mana $\overline{r_i = \frac{n}{n_i}}$ dan $s_i = r_i^{-1} \bmod n_i$ untuk i = 1,2,....,k.

Jika bilangan bulat $n_1, n_2, \ldots, n_k$ adalah pasangan relatif prima dan $n = n_1 n_2 \ldots n_k$, maka untuk semua bilangan bulat a, b pasti akan valid di mana a ≡ b mod n jika dan hanya jika a ≡ b mod n_i untuk setiap i = 1, 2,...,k..

Sebagai konsekuensi dari CRT, setiap bilangan bulat positif a < n dapat direpresentasikan secara unik sebagai sebuah k-tuple [$a_1, a_2, \ldots, a_k$] dan sebaliknya. Di mana a_i menunjukan sisa / residu a mod n_i untuk setiap i = 1, 2, …,k. Konversi a menjadi sistem residu didefinisikan dengan $n_1, n_2, \ldots, n_k$ dilakukan secara sederhana dengan reduksi modular a mod n^i. Konversi balik dari representasi sisa menjadi "angka-angka standar" adalah lebih sulit seperti yang dibutuhkan dalam kalkulasi pada persamaan di atas.

Kelebihan Chinese Remainder Theorem adalah sebagai berikut :

- Mempercepat untuk operasi kunci pribadi (dekripsi, pemberian tandatangan digital).
- Dua n/2-bit eksponensials mod P dan mod Q. sebagai ganti satu n-bit eksponensial mod N (N=P*Q).
- Split n-bit multiplier hardware ke dalam dua n/2-bit pengali, melaksanakan n/2-bit eksponensials paralel.
- Kombinasi hasil menurut CRT.
- CRT meningkat/kan decryption melewati suatu faktor aproksimasi 3- 3.5.

Kalkulasi penting di dalam rencana enkripsi RSA adalah eksponensial modular $M = E^d \pmod n$. Ini dilakukan setiap kali bagian dari pesan dilakukan enkripsi/dekripsi. d dan n adalah bilangan bulat yang sangat besar, oleh karena itu operasi ini sangat computationaly intensive. Sehingga harus ditemukan alternatif metoda biner untuk eksponensial modular.

Keuntungan dasar dengan menggunakan *Chinese Remainder Theorem* adalah memungkinkan untuk membagi modulo eksponensial yang besar ke dalam dua eksponensial yang jauh lebih kecil, satu di atas p dan satu di atas q. Dua modulo ini adalah faktor utama dari n yang dikenali. Kemudian masalah mengurangi ukuran dengan penggunaan teoreme Fermat's yang lebih kecil. Metoda ini pertama diusulkan oleh Quisquater dan Couvreur.

2.3. *Hensel Lifting*

Hensel Lifting adalah sebuah teknik yang digunakan untuk menyelesaikan suatu permasalahan kongruen polinomial bentuk $f(x) \equiv 0 \pmod{p^i}$, dimana p adalah bilangan prima. Misalkan kita mempunyai akar a dari sebuah f(x) modulo p^i. Kita ingin melakukan penyelesaian dari f(x) modulo p^{i+1} dari akar a yang telah kita dapat dimana akar dari mod p^{i+1} kongruen dengan a mod p^i. Akar tersebut akan mempunyai bentuk $a + np^i$ untuk beberapa nilai n. Sehingga kita ingin mencari nilai n agar $f(a + np^i) \equiv 0 \pmod{p^{i+1}}$.

Definition 1 *Untuk 2 elemen f, g ∈ R, dan I adalah sebuah ideal dari R, kita dapat katakan bahwa pseudo-PBB dari f, g adalah 1 (mod I) jika terdapat bilangan a, b ∈ R sehingga*

$$af + bg = 1 \pmod I$$

Lemma 2 *(Hensel's Lifting Lemma). Misalkan R adalah sebuah arbitrary commutative ring with identity dengan sebuah ideal I. Jika f ∈ R dapat ditulis sebagai f = gh (mod I) dimana pseudo-PBB dari g dan h adalah 1 (mod I), kita dapat mengankat faktorisasi ini dimana terdapat g' dan h' serta a', b' sehingga*

$$\begin{aligned} f &= g'h' \pmod{I^2} \\ a'g' + b'h' &= 1 \pmod{I^2} \\ g' &= g \pmod I \\ h' &= h \pmod I \end{aligned}$$

Dan lebih jauh, berikut ini juga berlaku:

- *Diberikan a, b, g, h, kita dapat secara mudah menghitung a', b', g', dan h'.*
- *Solusi g', h' adalah unik jika g" dan h" juga memenuhi persamaan tersebut, sehingga*

$$\begin{aligned} g'' &= g'(1+u) \pmod{I^2} \\ h'' &= h'(1-u) \pmod{I^2} \end{aligned}$$

Untuk beberapa u ∈ I.

Bukti. Definisikan g' = g + bm dan h' = h + am, dimana $f - gh = m \pmod I$. Sekarang,

$$\begin{aligned} f - g'h' &= f - (g + bm)(h + am) \\ &= f - gh + m(ag + bh) \pmod{I^2} \\ &= m(1 - (ag + bh)) \pmod{I^2} \\ &= 0 \pmod{I^2} \end{aligned}$$

Untuk pseudo-PBBnya, misalkan a' = a + am' dan b = b + bm' dimana m' = 1 − (ag' + bh') ∈ I.

$$\begin{aligned} a'g' + b'h' &= (a + am')g' + (b + bm')h' \pmod{I^2} \\ &= (ag' + bh') + m'(ag' + bh') \pmod{I^2} \\ &= 1 - m' + m'(1 - m') \pmod{I^2} \\ &= 1 - m' + m' - m'^2 \pmod{I^2} \\ &= 1 \pmod{I^2} \end{aligned}$$

Untuk keunikannya, misalkan g" dan h" juga memenuhi persamaan tersebut, dan sehingga g"− g' = $m_1 \in I$ dan h"− h' = $m_2 \in I$. Sehingga $u = m_1 a' - m_2 b' \in I$ karena kedua m_1 and m_2 berada dalam I.

$$
\begin{aligned}
f = g''h'' &= g'h' \pmod{I^2} \\
(g' + m_1)(h' + m_2) &= g'h' \pmod{I^2} \\
\implies m_1h' + m_2g' &= 0 \pmod{I^2} \\
\implies m_2g' &= -m_1h' \pmod{I^2} \\
a'm_2g' &= -a'm_1h' \pmod{I^2} \\
m_2(1 - b'h') &= -m_1a'h' \pmod{I^2} \\
m_2 &= h'(m_2b' - m_1a') \pmod{I^2} \\
m_2 &= h'(-u) \\
\implies h'' &= h'(1 - u)
\end{aligned}
$$

Mirip untuk g".

3. PEMBAHASAN

3.1. Modifikasi RSA

Untuk menggunakan *Chinese Remainder Theorem* dan *Hensel Lifting* dalam algoritma RSA, diperlukan beberapa perubahan pada struktur perpangkatannya yaitu pada nilai n. Jika pada algoritma RSA awal n = pq, maka pada modifikasi algoritma RSA nilai n menjadi $n = p^{b-1}q$. Dalam implementasinya, p dan q berukuran n/b bits. Apabila n berukuran 1024-bits, pada umumnya kita dapat menggunakan b = 3, sehingga $n = p^2q$. Kedua nilai prima p, q masing-masing berukuran 341 bits.

Algoritma modifikasi RSA:

1. Pilih dua bilangan prima sembarang, p dan q.
2. Hitung $n = p^2q$.
3. Hitung $\phi(n) = (p - 1)(q - 1)$.
4. Pilihlah kunci publik, e, yang relatif prima terhadap $\varphi(n)$.
5. Bangkitkan kunci privat $d = (1 + k\varphi(n)) / e$.
6. Hitung $r_1 = d \bmod (p - 1)$ dan $r_2 = d \bmod (q - 1)$.

Hasil dari algoritma di atas adalah:

- Kunci publik adalah (e,n)
- Kunci privat adalah (p, q, r_1, r_2)

Proses enkripsi dan dekripsi:

- Enkripsi
  1. Ambil kunci publik penerima pesan, e, dan modulus n.
  2. Nyatakan plainteks m menjadi blok-blok m_1, m_2, ..., sedemikian seterusnya sehingga setiap blok merepresentasikan nilai di dalam selang [0, n -1].
  3. Setiap blok m, dienkripsi menjadi blok c_i dengan rumus $c_i = m_i^e \bmod n$.
- Dekripsi
  1. Setiap blok chiperteks c_i didekripsi kembali menjadi blok m_i. Langkah-langkah berikutnya merupakan operasi tiap blok c_i menjadi m_i.
  2. Hitung $m_1 = c^{r1} \bmod p$ dan $m_2 = c^{r2} \bmod q$; sehingga $m_1^e = c \bmod p$ dan $m_2^e = c \bmod q$.
  3. Menggunakan *Hensel Lifting*, konstruksikan sebuah m_1' sehingga $(m_1')^e = c \bmod p^{b-1}$.
  4. Menggunakan *Chinese Remainder Theorem*, komputasikan sebuah $m \in \mathbb{Z}_n$ sehingga $m = m_1' \bmod p^{b-1}$ dan $m = m_2 \bmod q$. Kemudian $m = c^d \bmod n$.

Dilihat dari proses di atas, perbedaan modifikasi RSA ini dengan RSA biasa adalah penghitungan nilai n, kunci privat yang dihasilkan, dan proses dekripsi yang lebih rumit.

3.2 Performansi

Kita akan membandingkan usaha yang dilakukan proses dekripsi RSA biasa dengan RSA yang telah dimodifikasi dengan *Chinese Remainder Theorem* dan *Hensel Lifting*. Pada modifikasi RSA ini, proses dekripsi memerlukan 2 operasi perpangkatan modular yang penuh (n/b) bit dan (b-2)*Hensel Liftings*.

Sebuah operasi perpangkatan penuh adalah berukuran kubik dari modular sehingga perbandingan kecepatan modifikasi RSA ini dengan RSA biasa adalah:

$$\frac{2 \cdot (n/b)^3(b-2)}{2 \cdot (n/2)^3} = b^3(b-2)/8$$

Untuk 1024-bit RSA, nilai maksimal b yang didapat adalah 3, sehingga modifikasi RSA ini mempunyai kecepatan 3.38 lebih cepat dibandingkan dengan RSA biasa.

3.3 Keamanan

Keamanan dari modifikasi RSA ini tergantung dari kesulitannya dalam melakukan faktorisasi angka $n = p^{b-1}q$.

4. KESIMPULAN

Proses komputasi algoritma kriptografi kunci-publik RSA dapat dipercepat dengan melakukan modifikasi pada proses dekripsinya. Karena pada proses dekripsi terdapat operasi perpangkatan modular yang besar, yaitu nilai d.

Biasanya kunci publik e dari proses enkripsi RSA adalah nilai yang relatif rendah, contohnya 216 + 1 (sebuah nilai yang standar). Sehingga, pada proses enkripsi (bukan pada proses pemberian tanda tangan digital) tidak akan diperoleh masalah dengan kecepatan enkripsi karena bilangan pangkat e akan relatif kecil.

Chinese Remainder Theorem dan *Hensel Lifting* dapat meningkatkan kecepatan proses dekripsi RSA. Faktor peningkatan kecepatan yang dicapai adalah 3.38 (27/8). Selain meningkatkan kecepatan, modifikasi RSA ini juga dapat meningkatkan tingkat keamanannya, yaitu sulitnya melakukan faktorisasi dari operasi perpangkatan $n = p^{b-1}q$.

DAFTAR REFERENSI

[1] Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi,* Sekolah Teknik Elektro dan Informatika ITB, 2007.

[2] http://www.math.colostate.edu/~hulpke/lectures/m676ca/hensel.pdf, Akses: Desember 2007

[3] http://www.cse.iitk.ac.in/users/manindra/CS681/Lecture17.pdf, Akses: Desember 2007

[4] http://everything2.com/index.pl?node_id=1731805, Akses: Desember 2007

[5] T. Takagi. "Fast RSA-type Cryptosystem Modulo $p^k q$." In H. Krawczyk, ed., *Proceedings of Crypto 1998*, vol. 1462 of *LNCS*, pp. 318–326. Springer-Verlag, Aug. 1998.

[6] A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.