KLEPTOGRAFI PADA ALGORITMA KRIPTOGRAFI RSA

Albert Raditya S – NIM : 13506077

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16077@students.if.itb.ac.id

Abstrak

Kriptografi merupakan hal yang sangat penting dalam pengamanan data, terlebih dalam dunia telekomunikasi. Contoh-contoh aplikasi dari kriptografi adalah sebagai berikut: Algoritma AES, algoritma DES, dan sekuriti skema seperti SSH dan SSL. Karena semankin banyak orang yang menyadari pentingnya kriptografi, sehingga orang mencoba untuk mencari cara-cara untuk mendapatkan informasi yang disembunyikan dengan kriptografi. Salah satu cara untuk mendapatkan informasi tersebut adalah dengan menggunakan Kleptografi.

Kleptografi adalah sebuah studi untuk mencuri informasi secara aman dan rahasia. Kleptografi adalah ekstensi secara natural berdasarkan teori dari subliminal channels. Kleptografi diperkenalkan oleh Adam Young dan Moti Yung pada tahun 1996. Serangan Kleptografi adalah sebuah serangan *forward engineering* yang dibangun kedalam cryptosystem atau cryptographic protocol. Dengan kata lain, serangan kriptogfgrafi adalah sebuah teknik untuk menyerang kriptografi dengan metode kriptografi.

Pada makalah ini, penulis akan membahas beberapa aplikasi kleptografi pada algoritma RSA. Selain itu makalah ini juga akan membandingkan kleptografi dengan serangan-serangan terhadap kriptografi yang sejenis.

***Kata kunci**: Kriptografi, Kleptografi, Serangan, informasi, subliminal channels, RSA*

1 PENDAHULUAN

Informasi adalah salah satu hal yang sangat penting bagi manusia. Kita setiap hari berkomunikasi selain untuk berinteraksi dengan orang lain, kita juga berkomunikasi untuk mendapatkan informasi-informasi yang kita butuhkan. Seiring dengan perkembangan teknologi, komunikasi memang menjadi semakin mudah tetapi kebutuhan kerahasiaan informasi ketika berkomunikasi-pun semakin penting.

Karena itulah, mulai muncul metode-metode untuk mengamankan informasi. Tetapi keberadaan metode-metode untuk memberi pengamanan kepada informasi ini, juga menrik orang untuk mempelajari cara untuk mendapatkan informasi tanpa memiliki izin dari *authorize* user.

Walaupun keberadaan orang-orang yang berusaha memecahkan atau mendapatkan informasi dengan cara tersebut tidak baik, tetapi keberadaannya menyebabkan metode-metode kriptografi terus berkembang dan diperbaiki.

Selain itu kriptanalis juga berkembang untuk urusan militer dan negera, seperti untuk memecahkan kode yang dibuat oleh musuh.

2 TEORI DASAR

2.1 Kriptografi Kunci Public

Kriptografi kunci public atau Public key cryptography (lawan dari symmetric key cryptography) bekerja berdasarkan fungsi satu arah. Fungsi yang dapat dengan mudah dikalkulasi akan tetapi sangat sulit untuk dibalik/invers atau reverse tanpa informasi yang mendetail. Salah satu contoh adalah faktorisasi; biasanya akan sulit untuk memfaktorkan bilangan yang besar, akan tetapi mudah untuk melakukan faktorisasi. Contohnya, akan sangat sulit untuk memfaktorkan 4399 daripada memverifikasi bahwa 53 x 83 = 4399. Public key cryptography menggunakan sifat-sifat asimetrik ini untuk membuat fungsi satu arah, sebuah fungsi dimana semua orang dapat melakukan satu operasi (enkripsi atau verifikasi sign) akan tetapi sangat sulit untuk menginvers operasi (dekripsi atau membuat

sign) tanpa informasi yang selengkap-lengkapnya.

Public key cryptography dilakukan dengan menggabungkan secara kriptografi dua buah kunci yang berhubungan yang kita sebut sebagai pasangan kunci publik dan kunci privat. Kedua kunci tersebut dibuat pada waktu yang bersamaan dan berhubungan secara matematis. Secara matematis, kunci privat dibutuhkan untuk melakukan operasi invers terhadap kunci public dan kunci publik dibutuhkan untuk melakukan operasi invers terhadap operasi yang dilakukan oleh kunci privat.

Jika kunci publik didistribusikan secara luas, dan kunci privat disimpan di tempat yang tersembunyi maka akan diperoleh fungsi dari banyak ke satu. Semua orang dapat menggunakan kunci publik untuk melakukan operasi kriptografi akan tetapi hanya orang yang memegang kunci privat yang dapat melakukan invers terhadap data yang telah terenkripsi tersebut. Selain itu dapat juga diperoleh fungsi dari satu ke banyak, yaitu pada saat orang yang memegang kunci privat melakukan operasi enkripsi maka semua orang yang memiliki kunci publik dapat melakukan invers terhadap data hasil enkripsi tersebut.

2.2 RSA

RSA termasuk dalam kriptografi kunci public yang paling dikenal oleh orang banyak. RSA dirancang oleh 3 orang peneliti dari MIT pada tahun 1976. Oleh Ron Rivest, Adi Shamir, dan Leonard Adleman.

Seperti pada kriptografi kunci public, kekuatan RSA juga pada proses kalkulasi yang dibutuhkan untuk memecahkan hasil enkripsi, yaitu memfaktorkan bilanganyang besar menjadi factor-faktor prima.

Untuk mengenkripsi dan mendekripsikan dapat dirumuskan melalui

$$E(m) = c \equiv$$

$$D(c) = m \equiv$$

2.3 Kriptoanalis

Kriptoanalisis dapat diartikan sebagai seni atau ilmu untuk memecahkan cipherteks menjadi plainteks dengan memanfaatkan celah-celah keamanan sebuah sistem kriptografi. Hal inilah yang menjadikan kriptoanalisis dicap sebagai cara ilegal untuk menterjemahkan cipherteks. Orang yang melakukan kriptoanalisis disebut kriptoanalis, dan usaha untuk melakukan kriptoanalisis disebut dengan attack (serangan).

2.4 Kleptografi

Kleptografi merupakan sebuah seni dan aturan untuk mencuri informasi secara aman dan subliminal. Serangan Kleptografi memiliki basis terhadap desain black box cipher untuk membocorkan informasi kunci rahasia secara aman dan subliminal kepada desainer.

2.4.1 Kleptografi pada RSA

Untuk mencuri informasi pada RSA, diperlukan sebuah backdoor untuk mendapatkan kunci privat dari RSA yang dapat dideploy di pembangkitan kunci RSA sehingga

1. Backdoor hanya dapat diutilisasi oleh penyerang, walaupun kodenya didapatkan
2. Hasil pasangan kunci RSA harus sesuai dengan pasangan kunci yang normal
3. Kunci yang sama dari program pembangkitan kunci dimiliki oleh semua orang

Algoritma pembangkitan kunci akan dimodifikasi untuk berisi cryptotrojan, Cryptotrojan akan berisi kunci public penyerang, Y.

1. Pilihlah sebuah nilai besar s secara random
2. Hitung nilai p = H(s), dimana H adalah sebuah fungsi kriptografi satu arah
3. Jika p merupakan adalah komposit atau p-1 bukan relative prima dengan e maka kembali ke langkah pertama
4. Pilih nilai besar RND secara random
5. Hitung c agar enkripsi asimetri terhadap s dengan Y (public key dari attacker)
6. Selesaikan (q,r) pada (c || RND) = pq + r
7. Jika q adalah komposit ata q-1 tidak relative prima terhadap e maka kembali ke langkah pertama

Pengembalian Kunci Privat RSA

1. Penyerang mendapatkan nilai kunci public e dan n
2. Biarkan u sebagai 512 bits teratas dari n
3. Sang penyerang mengeset c1 = u dan c2 = u +1 (c2 untuk kemungkinan bit sisa yang terjadi karena komputasi n = pq = (c || RND) – r)
4. Penyerang mendekrip c1 dan c2 untuk medapatkan s1 dan s2

5. Antara p1 = H(s1) atau p2= H(s2) dapat membagi n

2.4 Serangan-serangan terhadap RSA

Terdapat beberapa serangan-serangan yang dapat dilakukan kepada algoritma RSA. Pada subbahasan ini akan dibahas 3 buah serangan yang dapat dilakukan terhadap RSA, yaitu factoring, H'asad Attack, dan Partial Key Exposure Attack.

2.4.1 Factoring

Serangan yang paling mudah dari RSA adalah faktorisasi dari modulus N = pq. Jika p telah diketahui, maka nilai q pun dapat dihitung, dan nilai φ(N) =N − p − q + 1 dapat diketahui. Dan hal ini cukup untuk menghitung d ≡ e -1 mod φ(N).

Saat ini metode paling cepat untuk memfaktorkan adalah General Number Field Sieve. Dengan kecepatan eksekusi

$$O\left(e^{(c+o(1))(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}}\right) = L_n\left[1/3, c\right]$$

2.4.2 H˚astad's Attack on Broadcasted Messages

Untuk mempercepat Enkripsi dan verifikasi dengan menggunakan RSA, akan mudah jika menggunakan public exponent e, misalnya 3. Tetapi cara seperti ini akan memaparkan RSA terhadap serangan berikut.

Semisal Bob ingin mengirimkan pesan M ke k penerima, yang mana semuanya menggunakan public exponent dengan nilai 3. Bob akan mendapatkan public key (N_i, e_i) untuk i = 1,... ,k. Dimana e_i = 3 untuk semua i. Kemudian dengan naive, Bob menghitung ciphertext C_i = M^3 mod N_i untuk semua i dan mengirimkan C_i ke penerima ke-i. Sebuah argumen sederhana akan menunjukkan bahwa setelah k ≥ 3, pesan M tidak lagi rahasia. Jika Evee mengintercept C_1, C_2, C_3, dimana Ci = M_3 mod N_i. Kita dapat mengasumsikan bahwa gcd(N_i,N_j) = 1 untuk semua i != j. Dengan menggunakan teori Chinese Remainder Problem, eve dapat menghitung C Z_{N1N2N3} yang mana C ≡ C_i mod N_i.Tetapi, karena M <Ni untuk semua i, kita mendapati bahwa M3 < N1N2N3. Sehingga C = M3 hold untuk semua integer, dan Eve dapat menhitung akar kubik dari c untuk mendapatkan M.

2.4.3 Timing Attack

Timing attack mengeksploitasi variansi dari operasi cryptographic. Jika RSA private key operasi dapat dihitung secara akurat, pada beberapa kasus statistik dapat diimplementasikan untuk mendapatkan kunci rahasia yang dibutuhkan untuk komputasi.

Untuk timing attack, penyerang harus memiliki beberapa kemungkinan nilai C^d mod N. Jika penyerang dengan teliti mengukur kebutuhan waktu yang dibutuhkan dan menganalisa variasi waktu, sang penyerang bisa mendapatkan kunci d satu bit persatu bit sampai keseluruhan d diketahui.

Serangan ini menurut Kocher adalah masalah pendeteksian signal. "Signal" ini terdiri dari variasi timing yang disebabkan oleh target exponent bit, sednagkan "noise" terdiri dari ketidakakuratan dalam penghitungan timing dan fluktuasi secara random.

```
x = C
for j = 1 to n
    x = mod(x^2, N)
    if d_j == 1 then
       x = mod(xC, N)
    end if
next j
return x
```

Algoritma Square and multiply

2.4.3 Partial Key Exposure Attack

Jika N = p.q adalah modulus RSA dan e,d adalah komponen enkripsi dan dekripsi maka ed ≡ 1 mod φ(N). Pada bagian ini penulis akan membahas mengenai seberapa banyak bit yang dibutuhkan dari d yang dibutuhkan untuk merekonstruksi d.

Hanya dibutuhkan seperempat dari least significant bit dari d, RSA lemah terhadap partial key exposure. Jika seorang musuh mencoba untuk menyerang system denagn berbagai cara dan ternyata dapat menghasilkan beberapa bit dari kunci, tetapi gagal untuk menemukan keseluruhan kunci. Serangan seperti timing attack, dapat dilakukan sampai seperempat least significant bit diketahui.Lalu musuh dapat menghitung dan menemukan d..

Partial Key Exposure Attack dapat diringkas menjadi dua buah teorema.

Teorema pertama: N = pq adalah n-bit RSA modulus dengan N = 3 mod 4. Dimana 1 ≤ e, d ≤ φ(N) memenuhi ed ≡ 1 mod φ(N) dan e <

2(n/4)−3. Ada algoritma dimana N, e dan n/4 bagian dari least significant bit dari d dan dapat mengkomputasi seluruh d secara polynomial n dan e.

Teorema Kedua N = pq adalah n-bit modulus RSA dengan N = 3 mod 4. $1 \leq e, d \leq \varphi(N)$ memenuhi $ed \equiv 1 \bmod \varphi(N)$.

- Jika e adalah prima dalam range {2t,...2t+1} dengann/4 ≤ t ≤ n/2. Maka ada algoritma yang jika diberikan N,e,dan t most significant bits dari d dapat mengkomputasikan semua d dalam waktu polynomial terhadap n.
- Secara lebih general ada e {2t,...,2t+1} adalah produk dari mayoritas prima dari r <e1,...er> dengan n/4 ≤ t ≤ n/2. Maka ada algoritma yang jika disediakan <e1,...er> dan t most significant bit dari d, dapat menkomputasikan d secara polynomial berdasarkan n dan 2^r.
- Ketika faktorisasi dari e tidak diketahui, kita bisa mendapatkan hasil yang lebih lemah. Untuk e pada range {2t,...,2t+1} dengan t 0 . . . n/2. untuk d > N untuk > 0. Terdapat algoritma yang jika diberikan N, e, , dan n-t most significant bit dari d dapat mengkomputasikan keseluruhan d dalam waktu polynomial n dan a/ .

3. Analisis dan Pengujian

3.1 Pengujian Algoritma Kleptografi

Berikut ini hasil pembangkitan nilai-nilai yang dibutuhkan sesuai dengan algoritma Kleptografi diats, dengan bantuan program kecil dengan seed e = 4, dan Y = 16

```
S:
100174916382561445247319056959611569283
056999343116218970894970436766512721700
205681654856664103753309206151704025845
146859963249649489775836674977811406 96

p: 830711494

RND:
760773032198723392558389659323668278334
111135771542998782624664819802309761116
766997637347084288937218117712566277558
922247258750859374942762862462158 9209

c:
200349832765122890494638113919223138566
113998686232437941789940873533025443400
411363309713328207506618412303408051690
293719926499298979551673349955622813 92

Temp:
268045681660743770459737797154406986082
539877000720970139132998730561391090511
641690243117099444737561561583067377285
458854079570675634728894331789277910 97

q:
322670004684856052394693117312768259448
857315318091614294111354537922634173292
950356412327550441642934052845869708509
725825557880959854311218103 58
```

Jika p dan q yang dihasilkan menggunakan algoritma diatas digunakan menjadi key dalam algoritma RSA, maka dengan mudah kunci rahasia dari pihak yang akan dapat diketahui berdasarkan algoritma recovery kunci privat pada teori diatas.

3.2 Analisis dan Perbandingan dengan algoritma sebanding

Pada bahasan ini, penulis akan membandingkan kleptografi dengan factoing, dan H'astad Attack.

Algoritma yang pasti dapat memecahkan kesulitan dalam mendapatkan informasi dari RSA adalah factoring. Factoring seperti layaknya algoritma bruteforce, dapat memecahkan dan menemukan kunci dengan memfaktorisasi n yang dipublish. Tetapi, karena tidak ada algoritma yang secara mangkus dapat memfaktorkan n, sehingga dibutuhkan waktu yang lama untuk memfaktorkannya

Algoritma H'astad's attack, pada H'astad's attack, juga memiliki banyak constraint yang mengikat yaitu pesan harus dikirimkan ke banyak orang sehingga orang yang menginginkan untuk mencuri informasi harus dapat memiliki akses ke ciphertext dari penerima yang cukup, sehingga dapat memecahkan kunci tersebut.

Algoritma H'astad Attack juga memiliki constraint pada penggunaan e yang terbatas. Oleh karena itu Penulis kurang menganjurkan penggunaan Algoritma H'astad's attack.

Sedangkan menurut penulis, sasaran algoritma kleptografi berbeda dengan dua algoritma yang penulis ulas sebelum ini, pada algoritma kleptografi harus disisipkan kedalam algoritma pembangkitan kunci pemiliki informasi yang ingin diambil informasinya oleh penyerang.

Selain factoring, algoritma timing attack dan partial key exposure juga sangat cocok untuk mencoba mencapatkan rahasia yang telah dienkripsi dengan algoritma RSA. Walaupun

algoritma timing attack membutuhkan statistic enkripsi yang dilakukan, tetapi timing attack cukup efektif dalam membobol dinding pertahanan yang ada dalam RSA.

Terlebih jika digunakan gabungan algoritma partial key exposure dengan timing attack, yang dapat mempersingkat waktu pencarian dari timing attack.

4. Kesimpulan

Dari perbandingan-perbandingan diatas dapat diketahui bahwa secara general algoritma RSA dapat dipecahkan dengan banyak cara. Walupun beluma ada yang efektif.

Secara general, untuk memecahkan permasalahan RSA penggunaan algoritma partial key exposure cukup baik untuk permaslaahn ini.

Kleptografi, juga merupakn cara yang cukup baik untuk mendapatkan informasi yang ada pada algoritma enkripsi RSA. Tetapi Penggunaan kleptografi harus disisipkan kedalam algoritma pembangkitan kunci RSA.

Daftar Pustaka

Munir, Ir. Rinaldi, M.T. Diktat Kuliah IF5054 Kriptografi. Teknik Informatika ITB, 2006

http://dimacs.rutgers.edu/Workshops/Intellectual/slides/yung.ppt

http://www.springerlink.com/content/qw9g86x7ktf8trum/

http://en.wikipedia.org/wiki/Kleptography

http://www.cryptovirology.com/cryptovfiles/cryptovirologyfaqver1.html

http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/E97/62.PDF

http://theory.stanford.edu/~gdurf/durfee-thesis-phd.pdf