Studi dan Implementasi *Optimal Asymmetric Encryption Padding*(OAEP) pada Algoritma RSA untuk Mencegah *Adaptive Chosen Ciphertext Attacks*

Tara Baskara – 135 04 042
Jurusan Teknik Informatika ITB, Bandung, email: if14042@students.if.itb.ac.id

Abstract – *Seiring meningkatnya perkembangan dunia teknologi, sistem pengamanan yang canggih terhadap suatu data semakin dibutuhkan. Hal ini juga didorong oleh semakin maraknya kejahatan di dunia maya. Salah satu sektor yang rawan mengundang kejahatan adalah sektor pengamanan data. Oleh karena itu, pengguna teknologi semakin beramai-ramai mengembangkan suatu sistem pengamanan terhadap data yang biasa disebut kriptografi. Dalam kriptografi kunci publik, terdapat beberapa algoritma yang sering dipakai, salah satunya adalah RSA. Akan tetapi berbagai serangan dapat dilakukan terhadap algoritma ini. Serangan-serangan tersebut tentunya akan menjadi ancaman untuk mengetahui kunci privat. Dengan memanfaatkan sedikit kesalahan, penyerang dapat mengetahui kunci privat dengan cara melakukan kalkulasi terhadapnya. Pada makalah ini akan dijelaskan bagaimana cara algoritma RSA menghadapi serangan adaptive chosen-ciphertext yang merupakan serangan paling kuat yang diketahui sampai saat ini, dengan menggunakan modifikasi OAEP.*

Kata Kunci*: RSA, OAEP, enkripsi, dekripsi, encode, decode.*

1 PENDAHULUAN

Sebagai makhluk sosial, komunikasi merupakan hal yang paling dekat dengan kita. Komunikasi dapat kita artikan sebagai berbagi pikiran, informasi dan intelijen. Segala bentuk aktivitas yang dilakukan oleh seseorang dengan tujuan menyampaikan pesannya pada orang lain merupakan tujuan komunikasi. Dilatarbelakangi oleh kebutuhan tersebut, manusia dapat melakukan pengiriman pesan dengan mudah dimana saja dan kapan saja dengan menggunakan berbagai media.

Perkembangan dunia digital saat ini membuat lalu lintas pengiriman pesan/data semakin pesat. Data yang dipertukarkan pun juga bervariasi baik dari jenisnya maupun tingkat kerahasiaannya. Mulai dari data pribadi, data organisasi sampai data negara yang sangat rahasia. Hal inilah yang menuntut adanya pengamanan data tersebut sehingga tidak sampai tersadap pihak ketiga.

Dalam kriptografi, enkripsi terhadap pesan digunakan agar kerahasiaan isi pesan dari siapapun yang tidak berhak untuk membaca pesan tersebut dapat terjaga. Yang dibahas dalam makalah ini adalah salah satu teknik dan algoritma kriptografi kunci publik yaitu RSA dengan modifikasi menggunakan OAEP untuk menghindari adaptive ciphertext-chosen attack.

2 DASAR TEORI

2.1 Kriptografi Kunci Publik

Pada kriptografi kunci publik setiap pengguna memiliki sepasang kunci, yaitu kunci publik dan kunci privat. Kunci untuk enkripsi diumumkan kepada publik, digunakan untuk enkripsi dan dilambangkan dengan *e*. Sedangkan kunci untuk dekripsi, bersifat rahasia, disebut kunci privat dan dilambangkan dengan *d*. Karena kunci enkripsi tidak sama dengan kunci dekripsi maka kriptografi kunci publik disebut pula kriptografi asimetri. Beberapa algoritma kriptografi kunci publik yang biasa digunakan adalah RSA, ElGamal, Schnorr, dan DSA.

2.2 RSA

RSA adalah salah satu 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. RSA dapat di implementasikan secara hardware dan software, dimana standar implementasi menggunakan PKCS#1. Pada perkembangannya RSA banyak digunakan karena kemudahan dan keamanannya.

2.2.1 Algoritma RSA

1. Pilih dua bilangan prima sembarang, *p* dan *q*
2. Hitung $n = p*q$(sebaiknya *p* tidak sama dengan *q*, sebab jika *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 $\phi(n)$
5. Bangkitkan kunci privat dengan menggunakan persamaan $e*d \equiv 1 \pmod{\phi(n)}$. Perhatikan bahwa $e*d \equiv 1 \pmod{\phi(n)}$ ekivalen dengan $e*d \equiv 1 + k\phi(n)$, sehingga secara sederhana *d* dapat dihitung dengan $d = (1 + k\phi(n)) / e$

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 kunci privat

2.2.2 Enkripsi/Dekripsi RSA

Enkripsi

1. Ambil kunci publik penerima pesan *e*, dan modulus *n*.
2. Nyatakan plainteks *m* menjadi blok-blok $m_1, m_2, \ldots$, sedemikian seterusnya sehingga setiap blok merepresentasikan nilai 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 cipherteks *ci* didekripsi kembali menjadi blok m_i dengan rumus $m_i = c_i^d \bmod n$

2.3 Cryptanalytic Attacks

Cryptanalytic attacks adalah usaha-usaha yang dilakukan seseorang untuk memperoleh informasi ataupun data yang telah dienkripsi. Tujuan *cryptanalytic attacks* adalah untuk mengetahui beberapa plainteks yang sesuai dengan cipherteks yang ada dan berusaha untuk menentukan kunci yang memetakan satu dengan yang lainnya. Plainteks ini dapat diketahui karena ia merupakan standar atau karena pendugaan. Jika suatu teks diduga berada di dalam suatu pesan, posisinya mungkin tidak diketahui, tetapi suatu pesan lazimnya cukup pendek sehingga memungkinkan kriptanalis menduga plainteks yang diketahui dalam setiap posisi yang mungkin dan melakukan penyerangan pada setiap kasus secara parallel. Salah satu dari *cryptanalytic attacks* ini adalah *adaptive chosen-text attack* yang akan dibahas dalam makalah ini.

2.4 Adaptive chosen-chipertext Attack

Adaptive chosen-ciphertext Attack adalah salah satu bentuk *cryptanalytic attack* dimana penyerang mengirimkan banyaknya cipherteks yang akan didekripsi, lalu menggunakan hasil dekripsi itu untuk memilih cipherteks berikutnya.

2.5 OAEP

Optimal Asymmetric Encryption Padding(*OAEP*) adalah suatu metode yang ditemukan oleh Mihir Bellare dan Phil Rogaway untuk melakukan *encoding* pesan lalu pesan di enkripsi menggunakan RSA.

OAEP melakukan *encoding* pesan yang mengandung "*masked data*" *string* digabung dengan "*masked random number*". Dalam bentuk sederhana, *masked data* terbentuk dengan melakukan XOR dari plainteks *M* dan hash *G* dari *random string r*. Sedangkan *masked random number* terbentuk dengan melakukan XOR dari *r* dengan *hash H* dari *masked data*. Lalu *input* pada fungsi enkripsi RSA adalah :

$$[M \text{ Å} G(r)] \parallel [r \text{ Å} H(M \text{ Å} G(r))]$$

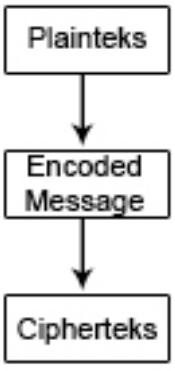
3 HASIL DAN PEMBAHASAN

3.1 Perancangan pembangunan aplikasi RSA-OAEP

Dalam membangun aplikasi RSA-OAEP, penulis melakukan langkah-langkah berikut :

1. Membuat fungsi *encode* OAEP
2. Membuat fungsi enkripsi RSA-OAEP
3. Membuat fungsi dekripsi RSA-OAEP
4. Membuat fungsi *decode* OAEP

3.2 Proses Enkripsi RSA-OAEP



Gambar 1. Proses enkripsi RSA-OAEP

3.2.1 Proses *Encoding* OAEP

Proses *encoding* dilakukan sebelum pesan di enkripsi dengan RSA. Proses itu meliputi beberapa tahap yaitu:

1. Membangkitkan *octet string PS*
2. Membangkitkan fungsi hash, dengan memasukkan parameter *encoding P* kedalam fungsi hash

   $$pHash = Hash(P)$$

3. Menggabungkan *pHash*, *PS*, pesan *M* dan padding lainnya untuk membentuk *data block DB.*

   $$DB = pHash||PS||01||M$$

4. Membangkitkan octet string acak *seed* dengan panjang *hLen*.
5. Memanggil fungsi pembangkit *mask MGF*, dengan panjang dari *octet* pesan, *emLen*.

   $$dbMask = MGF(seed, emLen - hLen)$$

6. Melakukan XOR antara *data block DB* dan *dbMask*

   $$maskedDB = DB \oplus dbMask$$

7. Memanggil fungsi pembangkit *mask MGF*

   $$seedMask = MGF(maskedDB, hLen)$$

8. Melakukan XOR antara *octet string* acak *seed* dengan *seedMask*

   $$maskedSeed = seed \oplus seedMask$$

9. Menetapkan $EM = maskedSeed||maskedDB$
10. Menghasilkan *EM*.

3.2.2 Proses Enkripsi

Setelah melakukan *encoding* terhadap pesan, akan dihasilkan *EM* yang kemudian akan di enkripsi. Berikut ini adalah tahap-tahap proses enkripsi :

1. Merubah pesan yang sudah di *encode EM* menjadi pesan dalam representatif integer *m* dengan fungsi primitif *OS2IP*.

   $$m = OS2IP(EM)$$

2. Memanggil fungsi enkripsi *RSAEP* dengan kunci publik (*n*, *e*) dan pesan *m* untuk menghasilkan cipherteks *c*

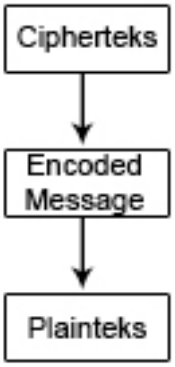
   $$c = RSAEP((n, e), m)$$

3. Merubah cipherteks *c* menjadi cipherteks *C* dengan panjang octet *k.*

   $$C = \text{I2OSP}(c, k)$$

4. Menghasilkan cipherteks *C*

3.3 Proses Dekripsi RSA-OAEP



Gambar 2. Proses dekripsi RSA-OAEP

3.3.1 Proses Dekripsi

Untuk mendapatkan pesan, maka pertama dilakukanlah proses dekripsi. Berikut ini adalah langkah-langkah proses dekripsi:

1. Cipherteks *C*, dirubah menjadi cipherteks berinterpretasi integer *c*
2. Memanggil fungsi dekripsi *RSADP* dengan kunci privat *K* untuk mendapatkan pesan *m*.

   $$m = RSADP(K,c)$$

3. Merubah pesan *m* menjadi pesan yang ter-*encode EM* dengan panjang k-1 *octet.*

   $$EM = I2OSP(m, \text{k-1})$$

3.3.2 Proses *Decoding* OAEP

Setelah melakukan dekripsi RSA terhadap pesan dan didapat cipherteks *EM*, maka dilakukan proses decoding untuk mendapatkan plainteks.

Proses decoding itu meliputi langkah-langkah berikut :

1. Menetapkan octet *hLen* pertama dari *EM* menjadi *maskedSeed* dan menetapkan *maskedDB* menjadi octet *emLen-hLen* sisanya
2. Memanggil fungsi pembangkit *mask MGF*

   $$seedMask = MGF(maskedDB, hLen)$$

3. Melakukan XOR antara *maskedSeed* dan *seedMask*

   $$seed = maskedSeed \oplus seedMask$$

4. Memanggil fungsi pembangkit *mask MGF*

$$dbMask = MGF(seed, emLen\text{-}hLen)$$

5. Melakukan XOR antara *maskedDB* dan *dbMask*

$$DB = maskedDB \oplus dbMask$$

6. Memanggil fungsi hash *Hash*

$$pHash = Hash(P)$$

7. Memisahkan *DB* menjadi *octet string pHash'* yang mengandung *octet hLen* yang pertama dari *DB, octet string PS,* dan pesan *M*

$$DB = pHash' || PS || 01 || M$$

8. Menghasilkan pesan *M*

3.4 Perbandingan RSA dengan OAEP dan RSA tanpa OAEP

Penyerangan *ciphertext adaptive* terhadap pesan yang terenkripsi RSA dan menggunakan PKCS #1 v1 *padding scheme*. Akan tetapi ada kecacatan pada skema PKCS#1 dan secara potensial semua kunci-kunci yang digunakan akan terungkap. Di lain pihak, serangan *ciphertext adaptive* mampu dipatahkan oleh RSA-OAEP.

3.5 Analisis Keamanan

Dalam sebuah *chosen-plaintext attack* (CPA), penyerang memiliki akses pada *sebuah encryption oracle*, yang berarti akses pada enkripsi seluruh plainteks yang dia miliki. Pada sebuah setting pada kriptografi kunci publik, hal ini tidak dapat dihindarkan. Sementara itu *adaptive chosen-ciphertext attacks* (CCA2) adalah sebuah skenario dimana penyerang menggunakan *decryption oracle* sebelum dan sesudah percobaan serangan. Satu-satunya batasan disini adalah bahwa penyerang tidak boleh mencoba oracle dengan cipherteks yang ingin ia pecahkan.

Adaptive chosen-ciphertext attacks diteliti secara teoritis sampai tahun 1998, ketika *Daniel Bleichenbacher* dari *Bell Laboratories* mendemonstrasikan penyerangan praktek melawan sistem yang menggunakan enkripsi RSA bersama dengan fungsi penyandi PKCS #1 v1, termasuk versi dari Protokol SSL(*Secure Socket Layer*) yang digunakan oleh ribuan *web server* pada saat itu. Pe nyerangan *Bleinchenbacher* mengambil keuntungan dari cacat yang terdapat pada fungsi PKCS #1 untuk membuka isi dari pesan RSA yang telah dienkripsi. Hal ini dilakukan dengan mengirimkan jutaan tes cipherteks ke alat pendekripsi (missal : *web server* yang dilengkapi dengan SSL).Pada prakteknya, ini berarti kunci sesi SSL dapat diketahui dalam waktu yang relatif singkat, mungkin satu hari atau kurang.

Tujuan dari penyerangan ini adalah untuk membuka informasi mengenai pesan yang telah dienkripsi atau mengenai kunci dekripsi itu sendiri. Untuk sistem dengan kunci publik, *adaptive chosen-ciphertext* umumnya dapat digunakan hanya ketika mereka mempunyai properti dari *ciphertext malleability*, yaitu cipherteks yang dapat dimodifikasi dengan suatu cara spesifik sehingga dapat mengakibatkan efek yang dapat diduga pada proses dekripsi pesan.

Dalam rangka mencegah *adaptive chosen-ciphertext attacks*, adalah suatu kewajiban untuk menggunakan pola enkripsi atau pengkodean yang dapat membatasi *ciphertext malleability*. Sejumlah pola pengkodean telah dikemukakan, dan yang paling umum untuk enkripsi RSA *adalah Optimal Asymmetric Encryption Padding* (OAEP). Tidak seperti pola pada PKCS #1 v1, OAEP telah dijamin aman terhadap model peramalan acak (*random oracle model*).

Optimal Asymmetric Encryption Padding(OAEP) adalah suatu metode yang ditemukan *oleh Mihir Bellare* dan *Phil Rogaway* untuk melakukan *encoding* pesan lalu pesan di enkripsi menggunakan RSA.

Lebih jauh lagi, *Bellare* dan *Rogaway* telah mengajukan sebuah konsep tentang kesadaran plainteks, dimana penyerang mencoba untuk memproduksi sebuah cipherteks yang valid tanpa mengetahui plainteks yang bersangkutan. Penekanan ini hanya didefinisikan pada *Random Oracle Model.*

Random Oracle Model diajukan *Bellare* dan *Rogaway* untuk menyediakan bukti-bukti heuristik mengenai keamanan yang sangat meyakinkan. Pada model ini, fungsi hash dianggap ideal, karena sifatnya yang acak. Dari sudut pandang sekuriti, hal ini mempengaruhi serangan dengan memberikan penyerang akses tambahan ke *random oracles* dari suatu skema.

4 KESIMPULAN

Dalam kriptografi, enkripsi terhadap pesan digunakan agar kerahasiaan isi pesan dari siapapun yang tidak berhak untuk membaca pesan tersebut dapat terjaga.

Pada RSA setiap pengguna memiliki sepasang kunci, yaitu kunci publik dan kunci privat. Kunci untuk enkripsi diumumkan kepada publik, digunakan untuk enkripsi. Sedangkan kunci untuk dekripsi, bersifat rahasia, disebut kunci privat. Maka dari itu, kunci privat harus dijaga agar tidak

disalahgunakan oleh orang lain.

Biasanya penyerang memanfaatkan kesalahan-kesalahan yang ada pada ciphertext atau tanda tangan untuk memecahkan kunci privat seseorang. Untuk menyerang algoritma RSA misalnya, penyerang dapat menyerang dengan *adaptive chosen-ciphertext attack*, yakni serangan yang paling kuat yang diketahui saat ini.

Untuk menangkal serangan *adaptive chosen-ciphertext attack*, dilakukanlah modifikasi pada RSA dengan menambahkan *Optimal Asymmetric Encryption Padding*(OAEP).

DAFTAR REFERENSI

[1] Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, 2006.

[2] Forouzan, Behrouz, *Cryptography and Network Security*, McGraw-Hill, 2008.

[3] http://en.wikipedia.org/wiki/RSA.Diakses pada bulan Desember 2007 – Januari 2008

[4] http://en.wikipedia.org/wiki/Adaptive_Chosen_Ciphertext_Attack.Diakses pada bulan Desember 2007 – Januari 2008

[5] http://en.wikipedia.org/wiki/Optimal_Asymmetric_Encryption_Padding. Diakses pada bulan Desember 2007 – Januari 2008

[6] RSA Laboratories.(2004). CryptoFAQ.http://rsasecurity.com. Diakses pada bulan Desember 2007 – Januari 2008

[7] http://en.wikipedia.org/wiki/Optimal_Asymmetric_Encryption_Padding.Diakses pada bulan Desember 2007 – Januari 2008