KRIPTOGRAFI DAN PEMANFAATANNYA PADA RSA

Tiffany Adriana – NIM 13505068

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if15068@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang kriptografi dan penggunaannya di berbagai bidang. Kriptografi merupakan ilmu dan seni yang mempelajari tentang cara menjaga kerahasiaan berita. Suatu berita dapat tetap terjaga kerahasiaannya dengan menggunakan suatu teknik yang disebut enkripsi. Melalui proses enkripsi, suatu berita dirubah menjadi bentuk lain yang tidak dapat langsung dibaca manusia maupun mesin. Dibutuhkan suatu kunci tertentu untuk dapat membaca berita yang telah dienkripsi. Proses ini dinamakan dekripsi.

Selain kriptografi secara umum, pada makalah ini akan dijelaskan mengenai beberapa algoritma dan fungsi yang memanfaatkan kriptografi. RSA yang merupakan algoritma pada enkripsi *pubic key* akan dijabarkan di sini. RSA merupakan salah satu algoritma yang paling maju dalam bidang kriptografi *public key*. RSA dipercaya dalam mengamankan dengan menggunakan kunci yang cukup panjang.

1. KRIPTOGRAFI

1.1. Defenisi

Secara umum, kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan berita. Kriptografi (*cryptography*) dilakukan oleh seorang kriptografer. Ada empat tujuan utama dari kriptografi:

1. Kerahasiaan (*confidentiality*)
   Kriptografi digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka / mengupas informasi yang telah disandi. Kerahasiaan dijaga dengan melakukan enkripsi (penyandian).
2. Keutuhan (*integrity*)
   Berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak. Bentuk-bentuk manipulasi yang dapat dilakukan antara lain penyisipan, penghapusan, dan pensubstitusian data lain ke dalam data yang sebenarnya.
3. Autentikasi
   Berhubungan dengan identifikasi / pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
4. Non-repudasi
   Atau nirpenyangkalan, adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman / terciptanya suatu informasi oleh yang mengirimkan / membuat.

1.2. Algoritma Kriptografi

Suatu pesan yang tidak disandikan disebut sebagai *plaintext* ataupun dapat disebut juga sebagai *cleartext*. Proses yang dilakukan untuk mengubah plaintext ke dalam ciphertext disebut *encryption* atau *enciphering*. Sedangkan proses untuk mengubah ciphertext kembali ke plaintext disebut *decryption* atau *deciphering*.

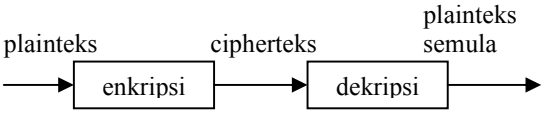
Algoritma kriptografi merupakan aturan untuk *enchipering* dan *dechipering*. Algoritma kriptografi dapat ditulis dalam suatu bentuk fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan, yaitu yang berisi elemen teks terang / plaintext dan yang berisi elemen teks sandi / ciphertext. Enkripsi dan dekripsi merupakan fungsi transformasi antara himpunan-himpunan tersebut. Apabila elemen-elemen plaintext dinotasikan dengan P, elemen-elemen ciphertext dinotasikan dengan C, sedang untuk proses enkripsi dinotasikan dengan E, dekripsi dengan notasi D, maka rumus matematis untuk enchipering dan dechipering dapat ditulis sebagai berikut:

Enkripsi : $E(P) = C$

Dekripsi : $D(C) = P$

atau

$D(E(P)) = P$

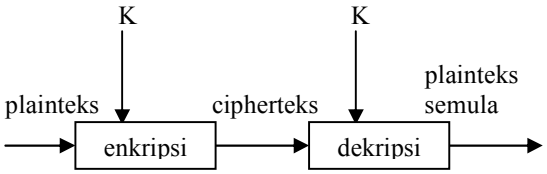


Kunci adalah parameter yang digunakan untuk transformasi *enciphering* dan *dechipering*. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara umum, kunci-kunci yang digunakan untuk proses pengenkripsian dan pendekripsian tidak perlu identik, tergantung pada sistem yang digunakan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi menjadi

$$E_K(P) = C$$
$$D_K(C) = P$$

dan kedua fungsi ini memenuhi

$$D_K(E_K(P)) = P$$



Dengan demikian keamanan suatu pesan tergantung pada kunci ataupun kunci-kunci yang digunakan, dan tidak tergantung pada algoritma yang digunakan. Sehingga algoritma-algoritma yang digunakan tersebut dapat dipublikasikan dan dianalisis, serta produk-produk yang menggunakan algoritma tersebut dapat diproduksi massal. Tidaklah menjadi masalah apabila seseorang mengetahui algoritma yang kita gunakan. Selama ia tidak mengetahui kunci yang dipakai, ia tetap tidak dapat membaca pesan.

Algoritma kriptografi harus memiliki kekuatan untuk melakukan:

- Konfusi / pembingungan (*confusion*), dari plaintext sehingga sulit untuk direkonstruksikan secara langsung tanpa menggunakan algoritma dekripsinya
- Difusi / peleburan (*difusion*), dari plaintext sehingga karakteristik dari plaintext tersebut hilang.

sehingga dapat digunakan untuk mengamankan informasi. Pada implementasinya, sebuah Algoritma Kriptografi harus memperhatikan kualitas layanan / *Quality of Service* atau QoS dari keseluruhan sistem dimana Algoritma Kriptografi tersebut diimplementasikan. Algoritma kriptografi yang handal adalah algoritma kriptografi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri.

1.3. Cryptographic Protokol

Suatu protokol adalah serangkaian langkah yang melibatkan dua pihak atau lebih dan dirancang untuk menyelesaikan suatu tugas.
Dari definisi ini dapat diambil beberapa arti sebagai berikut :

- protokol memiliki urutan dari awal hingga akhir
- setiap langkah harus dilaksanakan secara bergiliran
- suatu langkah tidak dapat dikerjakan bila langkah sebelumnya belum selesai
- diperlukan dua pihak atau lebih untuk melaksanakan protokol
- protokol harus mencapai suatu hasil

Selain itu, suatu protokol pun memiliki karakteristik yang lain, yaitu :

- setiap orang yang terlibat dalam protokol harus mengetahui terlebih dahulu mengenai protokol dan seluruh langkah yang akan dilaksanakan
- setiap orang yang terlibat dalam protokol harus menyetujui untuk mengikutinya
- protokol tidak boleh menimbulkan kerancuan
- protokol harus lengkap

Cryptographic protocol adalah suatu protokol yang menggunakan kriptografi. Protokol ini melibatkan sejumlah algoritma kriptografi, namun secara umum tujuan protokol lebih dari sekedar kerahasiaan. Pihak-pihak yang berpartisipasi mungkin saja ingin membagi sebagian rahasianya untuk menghitung sebuah nilai, menghasilkan urutan random, atau pun menandatangani kontrak secara bersamaan.

Penggunaan kriptografi dalam sebuah protokol terutama ditujukan untuk mencegah atau pun mendeteksi adanya *eavesdropping* dan *cheating*.

1.3.1. Fungsi Protokol

Dalam kehidupan kita sehari-hari terdapat banyak sekali protokol tidak resmi, misalnya saja dalam permainan kartu, pemungutan suara dalam pemilihan umum. Akan tetapi tidak ada seorang pun yang memikirkan mengenai protokol-protokol ini, protokol-protokol ini terus berkembang, semua orang mengetahui bagaimana menggunakannya.

Saat ini, semakin banyak interaksi antar manusia dilakukan melalui jaringan komputer. Komputer ini tentu saja memerlukan suatu protokol formal agar dapat melakukan hal yang biasa dilakukan manusia tanpa berpikir. Bila kita berpindah dari satu daerah ke daerah lain dan mengetahui bahwa kartu pemilihan suaranya berbeda dengan yang biasa kita gunakan, kita dapat beradaptasi dengan mudah. Akan tetapi kemampuan ini belum dimiliki oleh komputer, sehingga diperlukan suatu protokol.

Protokol digunakan untuk mengabtraksikan proses penyelesaian suatu tugas dari mekanisme yang digunakan. Protokol komunikasi adalah sama meskipun diimplementasikan pada PC atau VAX. Bila kita yakin bahwa kita memiliki protokol yang baik, kita dapat mengimplementasikannya dalam segala benda mulai dari telepon hingga pemanggang roti cerdas.

1.3.2. Penyerangan terhadap protokol

Penyerangan cryptographic dapat ditujukan pada beberapa hal berikut :

- algoritma cryptographic yang digunakan dalam protokol
- teknik cryptographic yang digunakan untuk mengimplementasikan algoritma dan protokol
- protokol itu sendiri

Seseorang dapat mencoba berbagai cara untuk menyerang suatu protokol. Mereka yang tidak terlibat dalam protokol dapat menyadap sebagian atau seluruh protokol. Tindakan ini disebut penyerangan pasif, karena si penyerang tidak mempengaruhi atau mengubah protokol, ia hanya mengamati protokol dan berusaha untuk memperoleh informasi.

Selain itu, seorang penyerang dapat berusaha untuk mengubah protokol demi keuntungannya sendiri. Ia dapat mengirimkan pesan dalam protokol, menghapus pesan, atau bahkan mengubah informasi yang ada di dalam suatu komputer. Tindakan-tindakan ini disebut sebagai penyerangan aktif, karena ia membutuhkan suatu campur tangan aktif.

Seorang penyerang tidaklah hanya berasal dari lingkungan luar protokol, namun ia mungkin juga berasal dari dalam protokol itu sendiri, ia dapat merupakan salah satu pihak yang terlibat dalam protokol. Tipe penyerang semacam ini disebut sebagai *cheater*. *Passive cheater* mengikuti protokol, tetapi berusaha memperoleh informasi lebih banyak daripada yang diperbolehkan protokol bagi dirinya. *Active cheater* mengubah protokol dalam usahanya untuk berbuat curang.

Usaha untuk menjaga keamanan protokol akan semakin sulit apabila pihak-pihak yang terlibat umumnya merupakan active cheater, oleh karena itu suatu protokol yang baik harus mampu atau pun harus aman terhadap kemungkinan *passive cheating*.

2. APLIKASI KRIPTOGRAFI

2.1. RSA

Dibidang kriptografi, **RSA** adalah sebuah algoritma pada enkripsi *public key*. RSA merupakan algoritma pertama yang cocok untuk *digital signature* seperti halnya enkripsi, dan salah satu yang paling maju dalam bidang kriptografi *public key*. RSA masih digunakan secara luas dalam protokol *electronic commerce*, dan dipercaya dalam mengamankan dengan menggunakan kunci yang cukup panjang.

2.1.1. Pembangkitan Kunci

Semisal **A** berkeinginan untuk mengijinkan **B** untuk mengirimkan kepadanya sebuah pesan pribadi (*private message*) melalui media transmisi yang tidak aman (*insecure*). **A** melakukan langkah-langkah berikut untuk membuat sebuah *public key* dan *private key*:

1. Pilih dua bilangan prima $p \neq q$ secara acak dan terpisah untuk tiap-tiap p dan q. Hitung $N = p\ q$. N hasil perkalian dari p dikalikan dengan q.
2. Hitung $\varphi = (p\text{-}1)(q\text{-}1)$.
3. Pilih bilangan bulat (*integer*) antara satu dan φ ($1 < e < \varphi$) yang juga merupakan coprime dari φ.
4. Hitung d hingga $d\ e \equiv 1 \pmod{\varphi}$.

- bilangan prima dapat diuji probabilitasnya menggunakan *Fermat's little theorem*- a^(n-1) mod n = 1 jika n adalah bilangan prima, diuji dengan beberapa nilai a menghasilkan kemungkinan yang tinggi bahwa n ialah bilangan prima. *Carmichael numbers* (angka-angka Carmichael) dapat melalui pengujian dari seluruh a, tetapi hal ini sangatlah langka
- langkah 3 dan 4 dapat dihasilkan dengan algoritma *extended Euclidean*; lihat juga aritmetika modular
- langkah 4 dapat dihasilkan dengan menemukan integer x sehingga $d = (x(p\text{-}1)(q\text{-}1) + 1)/e$ menghasilkan bilangan bulat, kemudian menggunakan nilai dari $d \pmod{(p\text{-}1)(q\text{-}1)}$

Pada *public key* terdiri atas:

- N, modulus yang digunakan.
- e, eksponen publik (sering juga disebut eksponen enkripsi).

Pada *private key* terdiri atas:

- N, modulus yang digunakan, digunakan pula pada *public key*.
- d, eksponen pribadi (sering juga disebut eksponen dekripsi), yang harus dijaga kerahasiaannya.

2.1.2. Proses enkripsi pesan

Misalkan **B** ingin mengirim pesan m ke **A**. **B** mengubah m menjadi angka $n < N$, menggunakan protokol yang sebelumnya telah disepakati dan dikenal sebagai *padding scheme*.

Maka **B** memiliki n dan mengetahui N dan e, yang telah diumumkan oleh **A**. **B** kemudian menghitung *ciphertext* c yang terkait pada n:

c = n^e \mod{N}

Perhitungan tersebut dapat diselesaikan dengan cepat menggunakan metode *exponentiation by squaring*. **B** kemudian mengirimkan c kepada **A**.

2.1.3. Proses dekripsi pesan

A menerima c dari **B**, dan mengetahui *private key* yang digunakan oleh **A** sendiri. **A** kemudian memulihkan n dari c dengan langkah-langkah berikut:

n = c^d mod{N}

Perhitungan diatas akan menghasilkan n, dengan begitu **A** dapat mengembalikan pesan semula m. Prosedur dekripsi bekerja karena

c^d ≡ (n^e)^d ≡ n^{ed} p mod{N}

Kemudian, dikarenakan $ed \equiv 1 \pmod{p\text{-}1}$ dan $ed \equiv 1 \pmod{q\text{-}1}$, hasil dari *Fermat's little theorem*.

n^{ed} ≡ n mod{p}
dan
n^{ed} ≡ n mod{q}

Dikarenakan p dan q merupakan bilangan prima yang berbeda, mengaplikasikan *Chinese remainder theorem* akan menghasilkan dua macam kongruen

n^{ed} ≡ n mod{pq}
serta
c^d ≡ n mod{N}

2.1.4. Pengesahan pesan

RSA dapat juga digunakan untuk mengesahkan sebuah pesan. Misalkan **A** ingin mengirim pesan kepada **B**. **A** membuat sebuah *hash value* dari pesan tersebut, di pangkatkan dengan bilangan *d* dibagi *N* (seperti halnya pada deskripsi pesan), dan melampirkannya sebagai "tanda tangan" pada pesan tersebut. Saat **B** menerima pesan yang telah "ditandatangani", **B** memangkatkan "tanda tangan" tersebut dengan bilangan *e* dibagi *N* (seperti halnya pada enkripsi pesan), dan membandingkannya dengan nilai hasil dari *hash value* dengan *hash value* pada pesan tersebut. Jika kedua cocok, maka **B** dapat mengetahui bahwa pemilik dari pesan tersebut adalah **A**, dan pesan pun tidak pernah diubah sepanjang pengiriman.

Harap dicatat bahwa *padding scheme* merupakan hal yang esensial untuk mengamankan pengesahan pesan seperti halnya pada enkripsi pesan, oleh karena itu kunci yang sama tidak digunakan pada proses enkripsi dan pengesahan.

2.1.5. Keamanan

Penyerangan yang paling umum pada RSA ialah pada penanganan masalah faktorisasi pada bilangan yang sangat besar. Apabila terdapat faktorisasi metode yang baru dan cepat telah dikembangkan, maka ada kemungkinan untuk membongkar RSA.

Pada tahun 2005, bilangan faktorisasi terbesar yang digunakan secara umum ialah sepanjang 663 bit, menggunakan metode distribusi mutakhir. Kunci RSA pada umumnya sepanjang 1024—2048 bit. Beberapa pakar meyakini bahwa kunci 1024-bit ada kemungkinan dipecahkan pada waktu dekat (hal ini masih dalam perdebatan), tetapi tidak ada seorangpun yang berpendapat kunci 2048-bit akan pecah pada masa depan yang terprediksi.

Semisal **E**, seorang *eavesdropper* (pencuri dengar—penguping), mendapatkan *public key* *N* dan *e*, dan ciphertext *c*. Bagimanapun juga, **E** tidak mampu untuk secara langsung memperoleh *d* yang dijaga kerahasiannya oleh **A**. Masalah untuk menemukan *n* seperti pada $n^e = c \bmod N$ di kenal sebagai permasalahan RSA.

Cara paling efektif yang ditempuh oleh **E** untuk memperoleh *n* dari *c* ialah dengan melakukan faktorisasi *N* kedalam *p* dan *q*, dengan tujuan untuk menghitung $(p-1)(q-1)$ yang dapat menghasilkan *d* dari *e*. Tidak ada metode waktu polinomial untuk melakukan faktorisasi pada bilangan bulat berukuran besar di komputer saat ini, tapi hal tersebut pun masih belum terbukti.

Masih belum ada bukti pula bahwa melakukan faktorisasi *N* adalah satu-satunya cara untuk memperoleh *n* dari *c*, tetapi tidak ditemukan adanya metode yang lebih mudah (setidaknya dari sepengatahuan publik).

Bagaimanapun juga, secara umum dianggap bahwa **E** telah kalah jika *N* berukuran sangat besar.

Jika *N* sepanjang 256-bit atau lebih pendek, *N* akan dapat difaktorisasi dalam beberapa jam pada Personal Computer, dengan menggunakan perangkat lunak yang tersedia secara bebas. Jika *N* sepanjang 512-bit atau lebih pendek, *N* akan dapat difaktorisasi dalam hitungan ratusan jam seperti pada tahun 1999. Secara teori, perangkat keras bernama TWIRL dan penjelasan dari Shamir dan Tromer pada tahun 2003 mengundang berbagai pertanyaan akan keamanan dari kunci 1024-bit. Santa disarankan bahwa *N* setidaknya sepanjang 2048-bit.

Pada thaun 1993, Peter Shor menerbitkan Algoritma Shor, menunjukkan bahwa sebuah komputer quantum secara prinsip dapat melakukan faktorisasi dalam waktu polinomial, mengurai RSA dan algoritma lainnya. Bagaimanapun juga, masih terdapat pedebatan dalam pembangunan komputer quantum secara prinsip.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Wikipedia. (2008). *http://id.wikipedia.org/wiki/Kriptografi*. Tanggal akses: 27 Desember 2008 pukul 20:00.

[3] Wikipedia. (2008). *http://id.wikipedia.org/wiki/RSA*. Tanggal akses: 27 Desember 2008 pukul 19:00.