PENERAPAN ALGORITMA KRIPTOGRAFI ASIMETRIS RSA UNTUK KEAMANAN DATA DI ORACLE

Ivan Wibowo [1] , Budi Susanto[2], Junius Karel T[3]

Abstrak:

RSA merupakan salah satu algoritma kriptografi asimetris yang menggunakan sepasang kunci, yaitu kunci publik dan kunci pribadi. Panjang kunci dapat diatur, dimana semakin panjang bit pembentukan kunci maka semakin sukar untuk dipecahkan karena sulitnya memfaktorkan dua bilangan yang sangat besar. Penelitian ini menerapkan algoritma RSA untuk enkripsi dan dekripsi kolom suatu tabel pada basis data Oracle sebagai basis data terbaik pada saat penulisan ini dibuat. Berdasarkan penelitian yang telah dilakukan, penulis dapat menarik kesimpulan, yaitu: Algoritma kriptografi asimetris RSA dapat diimplementasikan untuk proses enkripsi dan dekripsi kolom suatu tabel pada basis data Oracle, khususnya untuk tipe data *varchar2*.

Kata Kunci : *kriptografi, enkripsi, RSA, oracle.*

1. Pendahuluan

Untuk mengamankan data, salah satu cara dapat diterapkan suatu algoritma kriptografi untuk melakukan enkripsi. Dengan enkripsi data tidak dapat terbaca karena teks asli atau *plaintext* telah diubah ke teks yang tak terbaca atau disebut *chipertext*. Ada banyak algoritma kriptografi yang dapat digunakan, berdasarkan sifat kuncinya dibagi menjadi dua yaitu simetris yang hanya memakai satu kunci rahasia dan asimetris (*public key algorithm*) yang memakai sepasang kunci publik dan kunci rahasia.

Pada penelitian ini algoritma kriptografi yang akan digunakan adalah algoritma kriptografi asimetris RSA yang ditemukan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1978 dan RSA merupakan singkatan inisial dari nama mereka bertiga.

RSA digunakan karena merupakan algoritma kriptografi asimetris yang paling sering digunakan pada saat ini dikarenakan kehandalannya. Panjang kunci dalam bit dapat diatur, dengan semakin panjang bit maka semakin sukar untuk dipecahkan karena sulitnya memfaktorkan dua bilangan yang sangat besar tersebut, tetapi juga semakin lama pada proses dekripsinya. Melalui penelitian ini diharapkan RSA dapat diimplementasikan pada enkripsi dan dekripsi kolom tabel di Oracle sebagai salah satu basis data *server* terbaik pada saat penulisan ini dibuat.

2. Kriptografi

Teknik untuk mengacak suatu pesan agar tidak dapat diketahui maknanya disebut enkripsi, dan membentuk suatu bidang keilmuan yang disebut Kriptografi. Prinsip dasarnya adalah menyembunyikan informasi sedemikian rupa agar orang yang berhak saja yang dapat mengetahui isi dari informasi yang tersembunyi tersebut. Teknik ini sudah ada sejak jaman dahulu, bahkan sejak jaman sebelum Masehi pada masa perang yang digunakan untuk mengirim pesan rahasia antar sesama kawan agar apabila pesan terbaca oleh musuh ditengah jalan, isi dari pesan tersebut tidak dapat terbaca. Seiring dengan kemajuan teknik yang digunakan untuk mengenkripsi maka didalamnya terkandung unsur matematis yang membuat isi dari informasi itu semakin sulit untuk dibongkar.

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

---

[1] *Ivan Wibowo, Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana.Email : ivanoo46@gmail.com*

[2] *Budi Susanto, S.Kom., M.T., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*

([3] *Junius Karel T, M.T., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*

a) Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/ mengupas informasi yang telah disandi.
b) Integritas data, adalah 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, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
c) Autentikasi, adalah 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.
d) Non-repudiasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

3. Oracle

Oracle adalah basis data yang terdiri dari kumpulan data dalam suatu sistem manajemen basis data RDBMS. Perusahaan perangkat lunak Oracle memasarkan jenis basis data ini untuk bermacam-macam aplikasi yang bisa berjalan pada banyak jenis dan merk perangkat keras komputer (*platform*).

Basis data Oracle ini pertama kali dikembangkan oleh Larry Ellison, Bob Miner dan Ed Oates lewat perusahaan konsultasinya bernama *Software Development Laboratories (SDL*) pada tahun 1977. Pada tahun 1983, perusahaan ini berubah nama menjadi Oracle Corporation sampai sekarang.

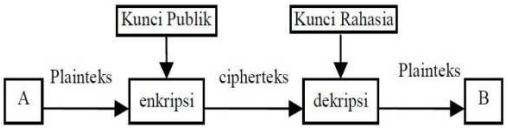
4. PL/SQL

PL/SQL *(Procedural Language/ Structured Query Language)* adalah bahasa gabungan antara bahasa prosedural dan bahasa SQL sehingga di dalamnya dapat dilakukan struktur kontrol seperti runtunan, percabangan dan perulangan. PL/SQL berada pada lingkungan *back end* dan digunakan untuk membuat kontrol terhadap *database* yang diolah sebelum *database* tersebut dihubungkan dengan lingkungan *front end.* Hal ini tentunya akan dapat menjaga integritas dan konsistensi data yang disimpan di dalamnya.

5. RSA

RSA merupakan algoritma kriptografi asimetris. Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman. Nama RSA sendiri diambil dari inisial nama depan ketiga penemunya tersebut. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci pribadi. Kunci publik boleh diketahui oleh siapa saja, dan digunakan untuk proses enkripsi. Sedangkan kunci pribadi hanya pihak - pihak tertentu saja yang boleh mengetahuinya, dan digunakan untuk proses dekripsi. Algoritma RSA masih digunakan hingga pada saat ini seperti yang diuraikan M. Zaki Riyanto dan Ardhi Ardhian:

Keamanan sandi RSA terletak pada sulitnya memfaktorkan bilangan yang besar. Sampai saat ini RSA masih dipercaya dan digunakan secara luas di internet. (*Kriptografi Kunci Publik: Sandi RSA*, 2008).



Gambar 1 Skema Kunci Asimetris

Skema algoritma kunci publik Sandi RSA terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Sebelumnya diberikan terlebih dahulu beberapa konsep perhitungan matematis yang digunakan RSA (*RSA and Public Key Criptography*, 2003, hlm 61).

Algoritma Pembentukan Kunci:

1. Tentukan p dan q bernilai dua bilangan Prima besar, acak dan dirahasiakan.
   $p \neq q$, p dan q memiliki ukuran sama.
2. Hitung $n = pq$
   Dan hitung $\phi(n) = (p-1)(q-1)$.
   Bilangan *integer* n disebut (*RSA*) *modulus*.
3. Tentukan e bilangan Prima acak, yang memiliki syarat:
   $1 < e < \phi(n)$
   $GCD(e, \phi(n)) = 1$, disebut e relatif prima terhadap $\phi(n)$,
   Bilangan integer e disebut (*RSA*) *enciphering exponent*.
4. Memakai algoritma Euclid yang diperluas (*Extended Eucledian Algorithm*).
   Menghitung bilangan khusus d,
   syarat $1 < d < \phi(n)$
   $d \equiv e^{-1} \bmod \phi(n)$
   $ed \equiv 1 \pmod{\phi(n)}$
   $ed \equiv 1 + k.\phi(n)$ untuk nilai k integer.

   Bilangan integer d disebut (*RSA*) *deciphering exponent*.
5. Nilai (n,e) adalah nilai yang boleh dipublikasi.
   Nilai d, p, q, $\phi(n)$ adalah nilai yang harus dirahasiakan.
   Pasangan (n,e) merupakan kunci publik.
   Pasangan (n,d) merupakan kunci rahasia.

Keterangan

- Fungsi $\phi(n)$ *Phi-Euler* merupakan fungsi terhadap bilangan bulat positif n yang meyatakan banyaknya elemen Zn yang mempunyai invers terhadap operasi pergandaan. Zn belum tentu merupakan grup terhadap operasi pergandaan, dengan kata lain, $\phi(n)$ adalah banyaknya elemen $\{x, 0 \leq x < n \mid gcd(x,n) = 1\}$
- Algoritma Euclid digunakan untuk mencari nilai GCD (Greatest Common Divisor) atau sering disebut FPB (Pembagi Persekutuan terbesar) dari dua bilangan bulat. Algoritma ini didasarkan pada pernyataan $gcd(r_0, r_1) = gcd(r_1, r_2) \ldots gcd(r_{n-1}, r_n) = gcd(r_n, 0) = r_n$

Contoh:
Akan dihitung gcd(40,24)
Jawab:

40 = 1.24 + 16	40 mod 24 = 16
24 = 1.16 + 8	24 mod16 = 8
16 = 2.8	16 mod 8 = 0, stop

Jadi gcd(40,24) = 8.
Dua buah bilangan bulat a dan b akan dapat dikatakan relatif prima jika $gcd(a,b) = 1$.

- Enkripsi: $c = m^e \bmod n$
- Dekripsi: $m = c^d \bmod n$

Contoh enkripsi:

Untuk mengenkripsi, dilakukan langkah – langkah sebagai berikut ini:

- Ubah tiap karakter teks terang menjadi bilangan bulat 01 - 26 (A = 01, B = 02, … , Z = 26), dan bagi teks menjadi beberapa blok b yang besar tiap bloknya lebih kecil dari n.
- Untuk tiap blok, hitung $c = b^e \pmod n$. c menjadi blok teks sandi yang dikirimkan.

Untuk mendekripkan kembali teks sandi, dilakukan langkah-langkah sebagai berikut :

- Hitung bilangan bulat d sedemikian hingga $de = 1 \pmod{(p-1)(q-1)}$. Pasangan (n, d) merupakan kunci rahasia.
- Untuk setiap blok sandi c yang diterima, hitung $b = c^d \pmod{n}$. Bagi pembuat sandi, dengan memilih 2 buah bilangan prima p dan q, tidaklah sulit untuk menghitung kunci publik $n = pq$, serta mendekripkannya kembali.

Contoh Perhitungan:

Andaikan B memilih $p = 13$ dan $q = 17$. Maka $n = pq = 221$.
Berikutnya, misalkan secara acak B memilih $e = 5$ yang merupakan bilangan yang relatif prima dengan $(p-1)(q-1) = 192$.
Maka kunci **publik** (n, e) = (221, 5).
A hendak mengirim teks "TAMAN", maka ia harus mengubahnya menjadi barisan angka - angka sebagai (A = 01, B = 02 , …): 20 01 13 01 14.
Misalkan A mengambil blok dengan panjang 3 digit, maka ia memiliki 4 blok untuk disandikan, masing - masing adalah 200, 113, 011, 4

- 200 disandikan menjadi $(200)^5 \pmod{221} = 200$
- 113 disandikan menjadi $(113)^5 \pmod{221} = 146$
- 011 disandikan menjadi $(11)^5 \pmod{221} = 163$
- 4 disandikan menjadi $(4)^5 \pmod{221} = 140$

B yang menerima pesan sandi dari A harus mencari kunci rahasia yang didapat dari relasi
$ed = 5d = 1 \pmod{192}$. Didapat $d = 77$.
Maka : didapat pesan asli 200 113 011 4 yang jika dikelompokkan dalam 2 digit menjadi 20 01 13 01 14 atau teks "TAMAN" seperti pesan semula.

- blok sandi 200 didekrip menjadi $(200)^{77} \pmod{221} = 200$
- blok sandi 146 didekrip menjadi $(146)^{77} \pmod{221} = 113$
- blok sandi 163 didekrip menjadi $(163)^{77} \pmod{221} = 11 = 011$ (karena 3 digit)
- blok sandi 140 didekrip menjadi $(140)^{77} \pmod{221} = 4$

6. Implementasi Proses Enkripsi Dekripsi

Karakter : **tugasakhir**
Kunci yang digunakan : **ivankey**

Pasangan Kunci Publik
(n, e) : (39917, 27017)

Pasangan Kunci Rahasia
(n, d) : (39917, 30689)

a. Proses mengubah karakter ke ASCII

Proses merubah karakter teks terang ke dalam bentuk ASCII menggunakan fungsi **fn_to_ascii**.
tugasakhir = 116 117 103 97 115 97 107 104 105 114

b. Proses Perhitungan Enkripsi

Proses perhitungan enkripsi dilakukan dengan fungsi **fn_enkrip**. Hasil deretan ASCII diatas kemudian dibagi per blok 4 digit untuk dilakukan perhitungan dengan metode RSA menggunakan pasangan nilai kunci publik, menjadi : 1161, 1710, 3971, 1597, 1071, 0410, 5114

Masing-masing angka ini kemudian dihitung menggunakan pasangan kunci publik.
Pasangan nilai kunci publik (n, e) : (39917, 27017)
Perhitungan enkripsi RSA:
$1161^{27017} \bmod 39917 = 36583$
$1710^{27017} \bmod 39917 = 10233$
$3971^{27017} \bmod 39917 = 30047$
$1597^{27017} \bmod 39917 = 21583$

1071^{27017} mod 39917 = 26887
0410^{27017} mod 39917 = 4285 = 04285
5114 = 5114

Agar pada saat proses perhitungan enkripsi dapat dijaga konsistensinya, maka setiap hasil tersebut harus dijadikan lima digit karakter dengan menambahkan karakter “0” didepannya. Pada hasil diatas 4285 menjadi 04285. Pada blok terakhir tidak dilakukan perhitungan, ini hanya algortima tambahan agar proses dekripsi dapat dilakukan dengan tepat. Kemudian hasil tersebut disatukan menjadi satu deretan karakter agar dapat diproses kebentuk karakter kembali.
Hasil deret: 365831023330047215832688704285 5114
Dengan hasil perhitungan tersebut maka karakter **“tugasakhir”** berhasil dienkripsi.
Enkripsi **“tugasakhir”** = 365831023330047215832688704285 5114

c. <u>Proses Perhitungan Dekripsi</u>

Proses perhitungan dekripsi dilakukan menggunakan fungsi **fn_dekrip**. Mengingat proses perhitungan enkripsi selalu dipaksa agar konsisten per blok memiliki lima digit angka, maka pada proses dekripsi ini, deret angka diatas dikelompokan per-blok lima digit, kecuali pada blok terakhir dibiarkan apa adanya, menjadi: 36583, 10233, 30047, 21583, 26887, 04285, 5114.

Masing-masing angka ini kemudian dihitung menggunakan pasangan kunci pribadi.
Pasangan nilai kunci pribadi (n, d) : (39917, 30689)
Perhitungan dekripsi RSA:
36583^{39917} mod 30689 = 1161
10233^{39917} mod 30689 = 1710
30047^{39917} mod 30689 = 3971
21583^{39917} mod 30689 = 1597
26887^{39917} mod 30689 = 1071
04285^{39917} mod 30689 = 410 = 0410
5114 = 5114

Untuk menjaga konsistensi proses dekripsi maka semua hasil per blok dijadikan empat digit dengan menambahkan karakter “0” didepannya, untuk nilai 410 menjadi 0410. Untuk blok terakhir tidak dilakukan proses perhitungan, karena pada proses enkripsi tidak dikenakan perhitungan. Kemudian hasil tersebut disatukan menjadi satu deretan karakter agar dapat diproses kebentuk karakter kembali.
Hasil deret: 116117103971159710710410 5114

d. <u>Proses Mengubah Deretan ASCII ke Karakter</u>

Proses ini menggunakan fungsi **fn_to_teks**. Hasil deret karakter diatas setelah proses perhitungan dekripsi RSA hasilnya adalah deretan ASCII yang akan diproses menjadi karakter kembali. Karena pada karakter yang digunakan untuk penelitian adalah “**tugasakhir**” dan semuanya adalah karakter pada keyboard yang tidak lebih dari ASCII 127 maka ada dua kemungkinan nilai ASCII, pertama ASCII yang lebih dari 100 dan ASCII yang kurang dari 100. Deretan tersebut dicek perbagiannya, jika ASCII lebih dari 100 dengan karakter awal “1” maka diambil per-blok tiga digit karakter dan jika ASCII kurang dari 100 diawali dengan karakter yang bukan “1” maka diambil per-blok dua digit. Sehingga deretan karakter diatas setelah dipotong-potong menjadi:
Deret : 116, 117, 103, 97, 115, 97, 107, 104, 105, 114
Kemudian dari potongan ASCII tersebut, masing-masing diproses ke karakter:

116 = t
117 = u
103 = g
97 = a
115 = s
97 = a
107 = k
104 = h
105 = i

114 = r
Hasil : **tugasakhir**

Karena hasil pengubahan ke karakter adalah **"tugasakhir"** maka hasil proses dekripsi adalah karakter **"tugasakhir"**.

Dekripsi 365831023330047215832688 7
042855114 = **"tugasakhir"**

7. Enkripsi Dekripsi menggunakan iSQL*Plus

Jika kolom belum diset trigger untuk enkripsi:
kunci: IVANKEY
password: 2c42e5cf1cdbafea04ed2670
18ef1511
(*password* hanya bisa dilihat *schema admin_user*)
INSERT INTO tusera1 values
('A03',admin_user.pkg_rsa.fn_enkrip
('tugasakhir','IVANKEY','2c42e5cf1cdbafea04ed267018ef1511'),'400','09-07-2009');
Jika kolom sudah diset trigger untuk enkripsi:
INSERT INTO tusera1
(no_ta1,name_ta1,qty_ta1,date_ta1) VALUES
('A02','tugasakhir','1000','30-06-2009');
Hasil enkripsi di iSQL*Plus:
365831023330047215832688 7042855
114

Dekripsi pada 1 kolom:
SELECT no_ta1,
admin_user.pkg_rsa.fn_dekrip(name_ta1,'IVANKEY','2c42e5cf1cdbafea04ed267018ef1511'),
qty_ta1, date_ta1
FROM tusera1 WHERE no_ta1 = 'A01';
Dekripsi pada banyak kolom sekaligus:
SELECT no_ta1,
admin_user.pkg_rsa.fn_dekrip(name_ta1,'IVANKEY','2c42e5cf1cdbafea04ed267018ef1511'),
qty_ta1, date_ta1
FROM tusera1 ORDER BY no_ta1;
Hasil dekripsi di iSQL*Plus: **tugasakhir**

8. Kesimpulan

Berdasarkan rumusan masalah, implementasi dan analisa sistem maka didapat kesimpulan sebagai berikut:

a. Algoritma kriptografi asimetris RSA (Rivest, Shamir, Adleman) dapat diterapkan pada basis data Oracle untuk menjaga keamanan data.
b. Algoritma kriptografi asimetris RSA (Rivest, Shamir, Adleman) dapat diterapkan pada tipe data scalar *varchar2*.
c. Fungsi pangkat number pada Oracle PL/SQL tidak dapat menangani pemangkatan terhadap bilangan yang sangat besar, sehingga digunakan kelas java untuk melakukan proses perhitungan.

9. Daftar Pustaka

Iqbal, Muhammad. 2006. *Studi Teknis Metode Enkripsi RSA dalam Perhitungannya*. Bandung: Institut Teknologi Bandung.

Menezes, Alfred J., Paul C. van Oorschot & Scott A. Vanstone. 2001. *Hand Book of Applied Cripthography*. CRC Press.

Mollin, Richard A. 2002. *RSA and PUBLIC-KEY CRYPTOGRAPHY*. Florida, Boca Raton: CRC Press LLC.

Mukodim, Didin. 2002. Teknik Pengamanan Data dengan RSA. *Proceedings, Komputer dan Sistem Intelijen (KOMMIT 2002)*. Jakarta: Universitas Gunadarma.

Riyanto, M. Zaki., & Ardhi Ardian. 2008. *Kriptografi Kunci Publik: Sandi RSA*. http://sandi.math.web.id

Setiawan, Deris. 2002. *Sistem Keamanan Komputer*. Jakarta: PT Elex Media Komputindo.

Wisesa, Swastyayana. 2008. *Metode Enkripsi RSA*. Bandung: Makalah Institut Teknologi Bandung.

Schneier, Bruce. 1996**.** *Applied Cryptography Second Edition*. John Wiley & Sons.