Studi dan Implementasi Pengamanan Basis Data dengan Teknik Kriptografi *Stream Cipher*

Dicky Ekklesia

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : dicky_ekklesia@yahoo.com

Abstrak

Masalah keamanan merupakan salah satu tantangan yang harus dipenuhi di dalam industri dan penelitian basis data. Data yang tersimpan di dalam basis data harus dapat terjamin keamanannya. Pengamanan data dapat dilakukan melalui dua cara. Cara pertama ialah pengaturan hak akses setiap pengguna oleh administrator basis data. Cara kedua ialah pengamanan data dari sisi kandungan data yang tersimpan pada basis data. Makalah ini menguraikan implementasi pengamanan data pada basis data dengan cara kedua. Pengamanan data dilakukan dengan menggunakan teknik kriptografi *RC4*. Penelitian (studi) yang dilakukan ialah untuk mencari cara agar *RC4* dapat dimanfaatkan untuk mengamankan data serta memberi kemudahan bagi pemilik data untuk mengamankan datanya tanpa perlu mengetahui *query – query* yang perlu diketikkan atau dijalankan.

***Kata kunci**: basis data, kriptografi, RC4*

1. Pendahuluan

Berbagai organisasi, perusahaan, atau pun pihak – pihak lain telah memanfaatkan teknologi basis data untuk menyimpan dan mengelola data organisasi atau perusahaannya. Saat ini, keamanan terhadap data yang tersimpan dalam basis data sudah menjadi persyaratan mutlak. Pengamanan terhadap jaringan komputer yang terhubung dengan basis data sudah tidak lagi menjamin keamanan data karena kebocoran data dapat disebabkan oleh "orang dalam" atau pihak – pihak yang langsung berhubungan dengan basis data seperti administrator basis data. Hal ini menyebabkan pengguna basis data harus menemukan cara untuk mengamankan data tanpa campur tangan administrator basis data. Kriptografi dapat digunakan untuk mengamankan data. Oleh karena itu, pengguna basis data membutuhkan bantuan untuk memenuhi kebutuhan keamanan akan data yang disimpannya.

Penerapan kriptografi pada makalah ini akan difokuskan bagaimana kriptografi dapat mengamankan data sampai pada level baris (*row*) dan kolom (*field*) dengan tetap memperhatikan integritas data dan kewenangan setiap pengguna basis data. Algoritma kriptografi yang akan digunakan ialah algoritma kriptografi simetris dan bersifat *stream cipher* sehingga data hasil enkripsi (cipherteks) mempunyai ukuran yang sama dengan data asli (plainteks). Teknik kriptografi simetris dipilih karena diharapkan dengan algoritma ini proses enkripsi – dekripsi data dapat dilakukan dengan waktu yang lebih cepat dibandingkan dengan algoritma kriptografi kunci publik (asimetris) [5].

2. Kriptografi

Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan (*message*) [5]. Algoritma kriptografi adalah [3] :

- Aturan untuk enkripsi (*enciphering*) dan dekripsi (*deciphering*).
- Fungsi matematika yang digunakan untuk enkripsi dan dekripsi.

Algoritma kriptografi berkembang terus dan terbagi atas dua bagian yaitu algoritma kriptografi klasik dan modern. Pada kriptografi klasik, kriptografer menggunakan algoritma sederhana, yang memungkinkan cipherteks dapat dipecahkan dengan mudah (melalui penggunaan statistik, terkaan, intuisi, dan sebagainya). Algoritma kriptografi modern dibuat sedemikian kompleks sehingga kriptanalis sangat sulit untuk memecahkan cipherteks tanpa mengetahui kunci. Algoritma kriptografi modern umumnya beroperasi dalam mode *bit*. Algoritma ini dapat dikelompokkan menjadi dua kategori yaitu cipher aliran (*stream cipher* – beroperasi dalam bentuk bit tunggal) dan cipher blok (*block cipher* – beroperasi dalam bentuk blok bit). Pengelompokan algoritma juga dilakukan berdasarkan kunci enkripsi – dekripsi yang digunakan, yaitu simetris (menggunakan kunci yang sama untuk proses enkripsi – dekripsi) dan asimetris atau kunci – publik (menggunakan kunci yang berbeda untuk proses enkripsi – dekripsi).

2.1 Algoritma Kriptografi RC4

Algoritma kriptografi *RC4* merupakan salah satu algoritma berjenis *stream cipher*. Algoritma ini akan memproses data dalam ukuran *byte* demi *byte* (1 *byte* = 8 *bit*). Algoritma ini dapat melakukan enkripsi dan dekripsi pada panjang data yang variabel atau dinamis tanpa perlu adanya penambahan *byte* (*padding*).

RC4 mempunyai sebuah *S-Box*, S0, S1, ... , S255, yang berisi permutasi dari bilangan 0 sampai 255, dan permutasi merupakan fungsi dari kunci K dengan panjang yang variabel. Langkah – langkah algoritma kriptografi *RC4* sebagai berikut :

1. Inisialisasi *S-Box*
   - isi *S-Box* secara berurutan, yaitu S0=0, S1=1, ... , S255=255.
   - Lakukan *padding* kunci K sehingga panjang kunci K = 256.
   - Lakukan pertukaran dan pengisian pada *S-Box* dengan kunci K, sebagai berikut :

     ```
     j = 0
     for i = 0 to 255
        j = (j + Si + Ki) mod 256
        swap Si dan Sj
     ```

     Fungsi swap merupakan fungsi yang menukarkan nilai S ke-i dengan nilai S ke-j

2. Proses enkripsi atau dekripsi *RC4* :

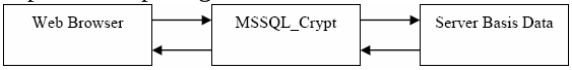
   ```
   i = 0
   j = 0
   for idx = 0 to len-1
       i = (i + 1) mod 256
       j = (j + Si) mod 256
       swap Si dan Sj
       t = (Si + Sj) mod 256
       k = St
       buffidx = k XOR buffidx
   ```

   Keterangan:
   - buff merupakan pesan yang akan dienkripsi atau dekripsi
   - len merupakan panjang dari buff

   Hasil akhir dari proses di atas ialah buff yang berisi pesan yang telah dienkripsi atau dekripsi.

3. Analisis

Perangkat lunak yang dikembangkan memberi penekanan pada kemudahan bagi pengguna umum basis data di dalam pemanfaatan perangkat lunak untuk melakukan pengamanan data melalui proses enkripsi dan dekripsi pada data yang dimiliki pengguna tersebut tanpa perlu melalui penulisan bahasa *query* tetapi cukup dengan melakukan "klik" pada tombol – tombol yang disediakan. Deskripsi umum sistem kerja perangkat lunak yang dibangun dapat dilihat pada gambar 1.



Gambar 1. Deskripsi Umum Sistem Kerja Perangkat Lunak

Perangkat lunak yang dibangun dinamakan **MSSQL_CRYPT**. Seperti terlihat pada gambar 1, perangkat lunak yang dibangun terletak di antara *server* basis data dan *web browser*. Hasil akhir dari perangkat lunak ini ialah aplikasi berbasis *web*. Pada dasarnya, perangkat lunak akan melakukan dua fungsi utama, yaitu :

1. Melakukan enkripsi – dekripsi data pada level kolom atau *field*.
2. Melakukan enkripsi – dekripsi data pada level baris atau *row*.

Perangkat lunak menerima masukan dari pengguna melalui antarmuka yang disediakan dan mengolah masukan dari pengguna tersebut serta mengubahnya menjadi *query* tertentu. *Query* ini kemudian dijalankan pada sistem basis data yang terdapat di *server*. Kemudian hasil dari *query* ini diberikan kembali kepada pengguna melalui antarmuka perangkat lunak.

Pengguna dari perangkat lunak **MSSQL_CRYPT** dapat dikelompokkan sebagai berikut :

1. Administrator basis data
   Administrator basis data merupakan pihak yang bertanggung jawab untuk melakukan pengaturan basis data antara lain: pembuatan basis data dan pengaturan pengguna basis data serta hak akses dari setiap pengguna tersebut.
2. Pemilik data
   Pemilik data merupakan pihak yang memiliki data yang terdapat pada basis data. Memiliki artinya pihak tersebut berhak melakukan pengubahan pada data seperti melakukan penambahan data baru, atau pun menghapus data, jadi bukan hanya memiliki hak untuk melihat isi dari data. Pemilik data merupakan pihak yang akan dan ingin melakukan perlindungan terhadap data yang dimilikinya dengan cara enkripsi maupun dekripsi terhadap data tersebut.

4. Perancangan

Perancangan merupakan proses pengolahan hasil analisis perangkat lunak menjadi rencana pengembangan perangkat lunak dan batasan – batasan perangkat lunak atau masalah yang mungkin dihadapi dalam pengembangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan arsitektur, perancangan modul, dan perancangan antarmuka.

4.1 Perancangan Arsitektur

Arsitektur perangkat lunak terdiri atas tiga buah proses utama yang dilakukan, yaitu :

1. Pemrosesan Otentikasi dan Kewenangan Data
   Otentikasi dan kewenangan data diproses berdasarkan nama pengguna (*user id*) dan *password* yang diterima sistem dan dicocokan

dengan data dan wewenang pada *server* basis data. Otentikasi merupakan pemrosesan wewenang pengguna untuk melakukan koneksi dengan *server* basis data sedangkan kewenangan data merupakan pemrosesan wewenang pengguna untuk melakukan manipulasi terhadap basis data dan tabel.

2. Pemrosesan *Query*
   Query – query yang diproses antara lain :
   a. Mendapatkan struktur tabel.
   b. Mendapatkan data yang tersimpan pada tabel.
   c. Mengubah data yang tersimpan.
   d. Mengubah struktur tabel.
3. Pemrosesan Pesan
   Pemrosesan pesan adalah proses untuk melakukan enkripsi – dekripsi pesan yang diterima berdasarkan kunci.

4.2 Perancangan Modul Perangkat Lunak

Perangkat lunak disusun atas lima modul utama, yaitu :

1. Modul Antarmuka
   Modul ini menyusun antarmuka perangkat lunak, menyediakan tempat dan *template* bagi modul lain untuk meletakkan hasil proses modul tersebut.
2. Modul Otentikasi
   Modul ini dirancang untuk implementasi dari proses konfirmasi login dan proses otentikasi pengguna.
3. Modul Pemrosesan *Query*
   Modul ini dirancang untuk implementasi pemrosesan *query*.
4. Modul Enkripsi – Dekripsi
   Modul ini dirancang untuk implementasi dari proses enkripsi – dekripsi.
5. Modul kriptografi *RC4*
   Modul ini dirancang untuk implementasi algoritma kriptografi *RC4*.

4.3 Perancangan Antarmuka

Antarmuka perangkat lunak terdiri atas dua halaman utama yaitu halaman login dan halaman aplikasi. Hasil dari implementasi perancangan antarmuka dapat dilihat pada bagian 5.

5. Implementasi

Hal – hal yang menjadi batasan pada tahap implementasi adalah :

1. Perangkat lunak hanya akan diimplementasikan pada satu macam sistem basis data relasional yaitu *Microsoft SQL Server 2000* (*MSSQL-Server*).
2. Hasil perancangan diimplementasikan pada komputer yang berfungsi sebagai *server* basis data dan *server* aplikasi *web*.

5.1 Implementasi Modul Kriptografi *RC4*

Modul ini terdiri atas tiga buah prosedur utama, yaitu :

1. Prosedur *RC4_swap*, untuk menukarkan nilai dua buah variabel.
2. Prosedur *RC4_PrepareKey*, untuk menyiapkan *S-Box* sebelum proses enkripsi atau dekripsi dilakukan.
3. Prosedur *RC4*, untuk melakukan enkripsi – dekripsi dengan algoritma *RC4*.

5.2 Implementasi Modul Otentikasi

Modul ini terdiri atas dua prosedur utama, yaitu :

1. Fungsi *connect*, untuk membuka koneksi dan melakukan otentikasi ke *MSSQL-Server*.
2. Prosedur *get_database_table*, untuk mendapatkan daftar basis data dan tabel pengguna (*user*).

5.3 Implementasi Modul Pemrosesan *Query*

Modul ini terdiri atas dua prosedur utama, yaitu :

1. Prosedur *view_structure*, untuk mendapatkan struktur dari suatu tabel.
2. Prosedur *view_data*, untuk mendapatkan data (baris) dari suatu tabel.

5.4 Implementasi Modul Enkripsi – Dekripsi

Modul ini terdiri atas empat fungsi utama, yaitu :

1. Fungsi *encrypt_col*, untuk melakukan enkripsi data secara kolom.
2. Fungsi *encrypt_row*, untuk melakukan enkripsi data secara baris.
3. Fungsi *decrypt_col*, untuk melakukan dekripsi secara kolom.
4. Fungsi *decrypt_row*, untuk melakukan dekripsi secara baris.

5.5 Implementasi Penanganan Integritas Data

Untuk menangani integritas tipe data, ada beberapa tipe data yang membutuhkan penanganan atau batasan tertentu, yaitu sebagai berikut :

1. Kelompok tipe data teks (*character strings*), terdiri atas : *char*, *varchar*, *text*, *nchar*, *nvarchar*, dan *ntext*.
   MSSQL_CRYPT tidak berhasil melakukan enkripsi maupun dekripsi pada tipe data *ntext*.
   Untuk tipe data teks lainnya, proses enkripsi dan dekripsi dapat berjalan dengan baik dengan syarat ukuran kolom mempunyai ukuran 33% lebih besar dari panjang data terbesar yang tersimpan pada kolom tersebut. Hal ini disebabkan karena proses enkripsi akan menghasilkan data yang setiap karakternya dapat bernilai keseluruhan kode *ASCII* yang ada sedangkan tipe data teks pada *MSSQL* tidak dapat menyimpan semua karakter *ASCII* sehingga perlu dilakukan konversi terhadap data

acak hasil enkripsi dengan cara melakukan pengkodean data tersebut secara *Base 64* (*Base64 Encoding*) terlebih dahulu sebelum dilakukan *update* (pengubahan) data pada basis data. Hasil dari pengkodean data secara *Base 64* ialah data dengan karakter yang dapat disimpan pada *field* dengan tipe data teks *MSSQL* dengan ukuran 33% lebih panjang dari data aslinya (data sebelum dikodekan).

2. Kelompok tipe data biner (*binary strings*), terdiri atas : *binary*, *varbinary*, dan *image*.
   Tidak ada penanganan dan batasan khusus untuk kelompok tipe data ini.
3. Kelompok tipe data numerik, terdiri atas : *bigint*, *int*, *smallint*, *tinyint*, *bit*, *decimal*, *numeric*, *money*, *smallmoney*, *float*, *real*, *datetime*, dan *smalldatetime*.
   MSSQL_CRYPT tidak melakukan enkripsi dan dekripsi pada tipe data *bit* karena ukuran data *bit* hanya satu *bit* sedangkan proses enkripsi dan dekripsi hanya dapat dilakukan pada data dengan ukuran minimal satu *byte*. Untuk mempertahankan tipe data pada setiap kolom, **MSSQL_CRYPT** akan melakukan enkripsi dekripsi tanpa mengubah tipe data tetapi data yang dienkripsi akan disimpan pada satu tabel tambahan sedangkan tabel yang mengalami enkripsi data, data bersangkutan akan diubah (*update*) menjadi nol. Tabel ini berfungsi sebagai tabel internal aplikasi **MSSQL_CRYPT** dan akan digunakan oleh semua data bertipe numerik yang akan dienkripsi. Tabel disimpan pada satu basis data internal dengan nama basis data "mcrypt" dan nama tabel "ud". Struktur dari tabel "ud" dapat dilihat pada gambar 2.

Column Name	Data Type	Length	Allow Nulls
rc4database	nvarchar	1000	NO
rc4table	nvarchar	1000	NO
pkval	nvarchar	128	YES
colname	nvarchar	128	YES
colval	nvarchar	4000	YES

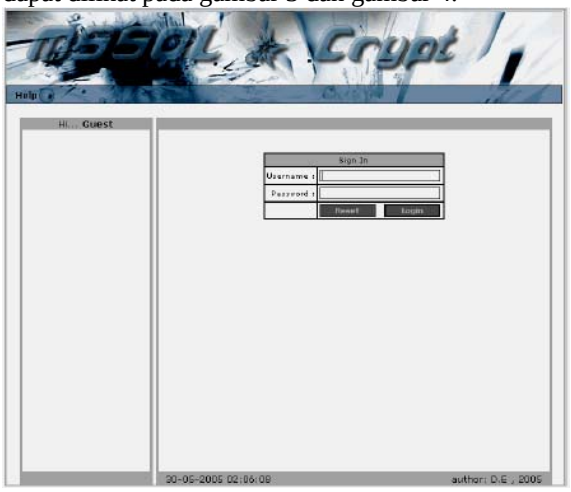
Gambar 2 Struktur Tabel ud (basis data internal "mcrypt")

4. Kelompok tipe data khusus, yaitu *cursor*, *sql_variant*, *table*, *timestamp*, dan *uniqueidentifier* serta tipe data yang didefinisikan oleh pengguna sendiri (*user defined type*).
   MSSQL_CRYPT tidak menangani kelompok tipe data ini. Penanganan kelompok tipe data ini memerlukan studi secara khusus. Selain itu, kelompok tipe data ini jarang digunakan.

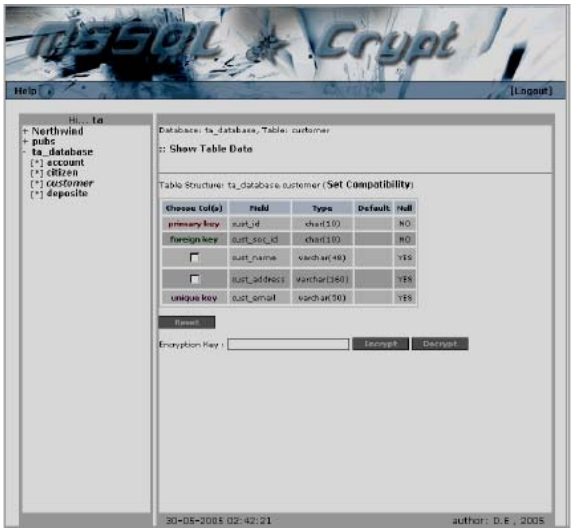
Selain penanganan integritas tipe data, **MSSQL_CRYPT** juga menangani *constraint* pada data. *Constraint* yang dimaksud adalah *primary key*, *foreign key*, dan *unique key*. **MSSQL_CRYPT** akan mengunci kolom yang memiliki salah satu *constraint* tersebut sehingga pengguna tidak dapat melakukan enkripsi ataupun dekripsi baik secara kolom maupun baris. **MSSQL_CRYPT** tidak menangani kolom yang memiliki *constraint check* karena *constraint check* dapat mengandung berbagai macam parameter sehingga sulit untuk diolah dan ditangani.

5.6 Implementasi Antar Muka

Hasil dari implementasi perancangan antarmuka dapat dilihat pada gambar 3 dan gambar 4.



Gambar 3 Tampilan Halaman Login



Gambar 4 Tampilan Halaman Aplikasi

6. Kesimpulan

Makalah ini membahas pemanfaatan kriptografi, khususnya teknik kriptografi *stream cipher RC4*, untuk pengamanan basis data. Beberapa kesimpulan yang dapat diambil dari pelaksanaan tugas akhir ini, yaitu :

1. Tipe data yang berhasil dienkripsi atau dekripsi dengan baik, antara lain :
   a. Kelompok tipe data teks (*character strings*), antara lain : *char*, *varchar*, *text*, *nchar*, dan *nvarchar*.
   b. Kelompok tipe data biner (*binary strings*), antara lain : *binary*, *varbinary*, dan *image*.

   c. Kelompok tipe data numerik, antara lain : *bigint*, *int*, *smallint*, *tinyint*, *decimal*, *numeric*, *money*, *smallmoney*, *float*, *real*, *datetime*, dan *smalldatetime*.
2. Tipe data yang tidak dienkripsi atau dekripsi, antara lain :
   a. Kelompok tipe data teks, antara lain : *ntext*.
   b. Kelompok tipe data numerik, antara lain : *bit.*
   c. Kelompok tipe data khusus, antara lain : *cursor*, *sql_variant*, *table*, *timestamp*, dan *uniqueidentifier* serta tipe data yang didefinisikan oleh pengguna sendiri (*user defined type*).
3. Batasan – batasan dari perangkat lunak yang dihasilkan, antara lain :
   a. Hanya dapat melakukan enkripsi dekripsi pada tabel yang mempunyai *primary key* dengan satu atribut.
   b. Tidak dapat melakukan pencegahan enkripsi dekripsi terhadap data yang memiliki *constraint check.*
   c. Tidak melakukan pengecekan terhadap kevalidan kunci enkripsi dekripsi yang digunakan.

7. Daftar Referensi

1. Ir. Fathansyah, *Basis Data*, Informatika, Bandung, 1999.
2. T. Marcus, A. Prijono dan J.Widiadhi, *DELPHI DEVELOPER dan SQL Server 2000*, Informatika, Bandung, 2004.
3. R. Munir, *Bahan Kuliah IF5054 Kriptografi*, Departemen Teknik Informatika, ITB, 2004.
4. A. Rahmani, *Implementasi Teknik Kriptografi Blowfish untuk Pengamanan Basis Data*, Tesis Magister Departemen Teknik Informatika, ITB, 2003.
5. B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition*, John Wiley & Sons, Inc, 1996.
6. A. Silberschatz, H. F. Korth. Dan S. Sudarshan, *Database System Concepts, 4th Edition*, McGraw – Hill, 2002.
7. B. Sukmawan, *RC4 Stream Cipher*, 1998.
8. B. Trower, *Crypt Data Packaging*, Trantor Standard Systems Inc, 2001.